PROTEUS / NEWS

AN INDEPENDENT NEWSLETTER FOR OWNERS AND USERS OF PROCESSOR TECHNOLOGY CORPORATION COMPUTERS

FORMERLY SOLUS NEWS

1982 - FIRST QUARTER

PUBLISHED BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA

VOL. 5 # 1
SINGLE ISSUE...$7.50 (US)
SINGLE ISSUE...$9.50 (FOREIGN)

McVIDEO UPGRADE AVAILABLE

IT REALLY WORKS!

by Stan Sokolow

In the last issue, I reported my preliminary experiences with the MicroComplex video upgrade for Sol. My Sol was a very early model (Rev D), so I expected it to be a tough test-case. I did report some problems, but just after the Proteus News went into the mail, I received the repaired unit and a phone call from Bob Hogg of MicroComplex. He explained that the early Sol's had a few timing problems, including slow 8080 microprocessors. He made a change to the video board to be more tolerant of these timing variations, and expected no further problems with mine. He was right. It works great.

At the Computer Faire, I exhibited my Sol with the video board in operation. People loved the crisp display it gives on the BMC monitor. They also liked the new features Bob has put into his version of Solos, which I call "McSolos" for "Micro-Complex Solos". (The "Mc..." was Bob's idea originally, calling his NorthStar DOS equivalent operating system "McDOS". I like it so much, I've given McNames to everything he makes. So we have McVideo, McFloppy, McSolos, and coming soon McWinchester. The names are kind of catchy, aren't they? I hope McDonald's Corporation doesn't have a copyright on all of them already.)

To review for those who didn't read the previous article, the McVideo upgrade is a circuit board custom made for the Sol. It is not an S-100 board and does not use any of the five S-100 slots in the Sol. Primarily, it converts the Sol's video display so that it can be switched between the usual 16x64 display format and the more standard 24x80 format. I say "primarily" because it does a lot more than that.

Other features include: (1) a sonic alert can be sounded under software control (it beeps); (2) the upgrade eliminates the annoying screen snow that was due to the video being switched off momentarily during update of the video RAM; (3) it allows the Sol's dedicated address space (ROM, RAM, video RAM) to be disabled under software control so that you can have a full 64K of RAM in the S-100 bus; (4) two latched output bits are available for custom use, such as disabling other memory mapped boards in the system; (5) it relocates Sol's address space from C000 to F000 when switched into the 24x80 mode; (6) it replaces Sol's slow RAM with very fast RAM for compatibility with future upgrades that speed up the Sol; and (7) the character generator is not changed, so the display still gives the same high-quality character font.

The video display monitor you use now may have to be adjusted or replaced with a better quality unit to provide the crisp characters this board is capable of producing. That may be an extra expense, but don't forget that you sit and stare at that tube all the time you use the computer. Your eyes and nerves deserve a good quality display.

To make full use of the board, you should use it in

CON'T ON PAGE 2

NEW VERSATILE DISK CONTROLLER: FLOPPY, HARD DISK, CLOCK IN ONE BOARD

There is a new disk controller on the market which offers high performance in one S-100 board compatible with Sol. Not only that, but it is plug compatible with the PerSci disk used in the Helios system. Named the "Versatile Floppy Disk Controller" (VFDC), this board is so good for Sol users, that Proteus has arranged to be a dealer for it.

Its use is not limited to the Helios, but Helios owners will rejoice to know that this board is a no-hassle plug-in replacement for the Helios controller board set that will give IBM compatible soft-sectoring instead of the unique Helios format. This will let Helios owners run CP/M 2.2 and buy software from virtually any source, instead of being dependent upon the limited software available from Lifeboat Associates on Helios CP/M.

This board provides a dual density 8" floppy disk controller, a high-speed parallel interface for an intelligent Winchester drive, and a calendar clock module. All of the functions use I/O ports rather than memory mapped addresses, so the board uses no address space. Both the hard disk and floppy access the system memory through Direct Memory Access (DMA) so that the Sol can continue to do useful work while the controller transfers data to and from memory. The calendar clock maintains date and time-of-day down to thousandths of seconds, and it continues to function with an onboard battery when the computer system is turned off.

The floppy disk controller portion of the board uses the Western Digital dual density floppy disk contoller chip, which means that it is able to perform the fast-seek with the PerSci drives. Many controllers on the market now use the NEC chip or equivalent, which is an elegant floppy disk controller but it can only step the drive at 1 ms maximum rate. The PerSci will accept step pulses in a rapid burst to do the fast-seek, but the NEC chip can't do that. The WD chip can. It provides IBM compatible, soft-sectored format (IBM 3740 single-density format, IBM system 34 double-density format).

Moreover, the controller and software know all about the PerSci drive's unique features. Many controllers are totally confused by the fact that the PerSci drive has both heads ganged together on one seek mechanism. (That is, when drive 0 seeks, so does drive 1.) The cable pinouts of the PerSci are non-standard, so most manufacturers do not support the PerSci directly, requiring cuts and jumpers on the controller and/or the drive pcb to swap the signals around. This VFDC controller has a configuration plug which adapts the board to the drive pinouts with no cutting. The adapter plug can be set up for any kind of drive. The special PerSci signals for Seek Complete, Spindle Motor Enable, and Direct Head Load are all supported by the board.

CON'T ON PAGE 2

McVIDEO UPGRADE - CON'T FROM PAGE 1

conjunction with the MicroComplex Dual Personality Module. Together, these two boards will provide the hardware and software to let you run the Sol with modern software needing a 24x80 screen, and still revert to the old Sol mode for running your old software.

Use of the 24x80 mode is easy if you treat Sol as a console device through calls to McSolos. All of the standard entry offsets have been preserved. Software that uses the memory-mapped video directly, however, will need modifications to know the new origin (FC00) and size of the display RAM. I have quickly patched PTDOS's console driver to use the F-origin entry points, but haven't had time to modify the PTDOS memory-mapped EDIT. The navigational editor (the one that "yanks" text as a stream) should work fine until EDIT is altered. If someone wants to work on EDIT, I'll be happy to send the source code if they don't have it already. I'm sorry to say that modifying WordWizard will not be possible, so it will have to be used in the 16x64 mode.

I am completely satisfied with the McVideo board and am sure you will be, too. Proteus has arranged to be a dealer for the MicroComplex products, so that we can derive some needed income from the sales. Factory service will still be available for any problem we can't handle, but Proteus will try to take some of the work off of Bob's shoulders so he can get on with his designs for more Sol upgrades. He's working on a Z-80 board, which should be available by the time of the next Proteus News. We'll have to call it "McZol", of course.

Bob's also thinking about an 8088 board, but other things have to come first. He wants to wait until the IBM'ers have developed lots of 8088 software anyway. I have noticed that 8088/8086 versions of the same ol' software available already for CP/M-80 cost about twice as much right now.

The Dual Personality Module, programmed with the new McSolos, is available from Proteus for $95. The McVideo board is $295 through Proteus, fully assembled. Not a bad price when you consider that it makes the Sol over in so many ways. And I guarantee you'll be happy with it. If not, let me know why and return it in good condition within 30 days for a full refund of your purchase price.

Consult the Proteus catalog in this issue for more details.

NEW VERSATILE DISK - CON'T FROM PAGE 1

The hard disk interface circuitry is designed to connect directly to the PRIAM "SMART" interface which controls up to four PRIAM disk drives of 10 to 158 megabytes each. You can now give your Sol or S-100 computer truly mass storage in one S-100 slot. The hard disk uses DMA for access to system RAM, so I/O can overlap processing.

The calendar clock uses the National Semiconductor MM58167 chip which provides time and date in digital form. That is, the following digits can be read: thousandths of seconds, hundredths of seconds, tenths of seconds, seconds, minutes, hours, day of week, day of month, and month. The onboard battery provides approximately one year between battery changes under normal use. When the system is turned on, the clock chip derives power from the computer; but when turned off, the battery continues the clock without interruption.

Having date and time available gives your system new power. Here are some examples. Your software can reliably determine the time by reading the clock using the clock utility program. This is very useful for "date-and-time-stamping" your files when they are changed, so that you can be certain of which file is the latest. Programmers will find this handy for keeping source code revisions in order during frequent changes as debugging is taking place. (A video-oriented editor which can read the date and time into the edited file is available for CP/M with this board.)

Security break-in attempts in business systems can be logged with date and time for later investigation. That is, it is not good enough just to provide password protection to keep non-authorized users out of files they shouldn't access. You also need to be aware of the attempt to breach the security of the system so the culprit can be found. This is a key technique in discovering embezzlement attempts in financial systems.

The clock would also allow high-security systems to have an automatic lock-out based upon a timer routine, so that someone trying to discover a password by running through all possible permutations will be slowed down to one attempt every 10 minutes or something like that, making this exhaustive search method truly exhausting.

Using a log-file, your system can keep track of the date and time of hardware errors that give early warning of malfunctions. For example, rather than just retrying a disk read that fails, why not log the date, time, disk drive, and diskette identification onto a trouble log file. Periodically you can run a program which summarizes the trouble log so you can look into why the disk is getting flakey before it fails completely. With some imagination, I'm sure users will come up with more applications for the calendar clock.

The VFDC board uses I/O ports A0 - BF and E0 - EF hex. These port addresses do not conflict with the Sol or Helios ports (which use F0 - FF), so the board can be used in a system with a Helios controller as well. To install the configured board in a Sol instead of the Helios boards, all you do is remove the Helios boards from the Sol, unplug the drive cable from the Helios controller, plug it onto the new controller, and put the controller into a bus slot of the Sol. No changes are necessary inside the drive, so at any time you can put the Helios controller back into the system.

However, you should be clear on one point. The Helios format disks cannot be written with the VFDC. Processor Tech used a unique recording format that cannot be used by other controllers on the market. It may be possible to read Helios disks with a special transfer program using the VFDC, but this is not yet clear. So, file transfer from the Helios format to the IBM format must be done by connecting two drives to the same computer (one with Helios controller and the other with the IBM compatible controller) or by connecting two different machines and passing data .by serial or parallel port.

A number of Proteus members have the Tarbell single-density controller and Helios controller connected to the same drive using a multiplexer feature of the old Tarbell board. They are able to pass data from the Helios format to the single-density IBM format using different slots in the same drive. I'm sure that members who want to convert old files to the new format will be able to arrange something with these people. Proteus will provide a referral service for this. Anyone interested in doing this data transfer for a reasonable fee should contact me to get on our list.

Of course, files can be moved from one format to the other via cassette tape. Richard .Greenlaw's TAPEDISK and DISKTAPE utilities move CP/M files to and from Sol's cassette tape. Our tape librarian Lewis Moseley can provide you with these utilities.

The board does not provide an automatic bootload ROM, so you must have a bootload routine (about 20 to 30 bytes) in ROM

CON'T ON PAGE 3

NEW VERSATILE DISK - CON'T FROM PAGE 2

somewhere. A modified version of Solos in 2708 or 2716 EPROMs, having both the Helios bootload and the VFDC controller's bootload routines, is available too.

Now about price. The assembled and tested VFDC board with a 70 page manual and a diskette having CP/M BIOS source code, disk formatting and copying utilities, disk diagnostic routines, and clock utility, sells for $495. A kit version, which also includes an assembly manual, sells for $395. A ready-to-boot CP/M 2.2 for VFDC in a standard Sol is $170. Included with the CP/M are a diskcheck command (similar to the PTDOS DCHECK), file compare, and disk recovery programs.

The modified bootload-Solos in EPROM (2708 or 2716) is $20. If you have one of the old personality modules that does not take the 2708, 2716, or 9216 ROM Solos, you will need to get another personality module. We recommend the dual personality module made by MicroComplex, for $95. This has been described in previous issues of Proteus News. It is available from Proteus also. When ordering it, specify that you want the bootload for Helios and VFDC. This personality module has the advantage that it is compatible with the MicroComplex video upgrade to 24x80 screen and 64K RAM.

The CP/M 2.2 BIOS for the VFDC in an upgraded Sol, with the 24x80 display, will be available soon. Purchasers of the VFDC or 24x80 display from Proteus will be able to get the BIOS for only mailing and diskette costs as soon as it is ready. The BIOS allows the upgraded Sol to work with 64K contiguous memory by switching out the Sol's dedicated address space when not in use.

If there is sufficient interest, I will put together a kit consisting of the necessary components to upgrade a two-slot Helios drive to contain a Priam hard disk in to empty space inside the Helios cabinet, along with the required power supply components, cables, controller, etc. The price of a 35 megabyte hard disk kit would be about $5000. Also available are 10 or 70 megabyte versions. Let me know if you are interested.

65K RAM Memory Modification For The Sol-20

Reprinted from "Microsystems" magazine.

by Jim Spann

Don't give your Sol to the trash man; a savior is here! This simple modification gives the Sol Terminal Computer* a new lease on life by moving the VDM* and Solos* memory to the Sol's internal data bus (where it belonged anyway), so it no longer interferes with the S-100 memory address space. Now you can run all those big memory programs without having to buy a new computer, and still have access to all the Solos utility routines. And all old programs will execute properly.

A minor wiring change and the addition of two integrated circuits is required to extend the usability of the Sol Terminal Computer. This feat is accomplished by the use of a memory management flip-flop, controlled by a unused output port in the Sol. The parts required are a SN74LS74 (flip-flop) and SN7406 (open collector inverter).

The required circuit changes can be made without cutting any printed circuit board traces. The technique of hanging the IC pin to be changed outside of the socket and soldering a wire to it can save much wear and tear on the circuit board (Figure 1). The two chip memory management control circuit can be assembled on a small vector board and mounted inside the Sol under the keyboard as shown in Figure 2.

The following steps refer to the Sol schematics and drawings in the Sol manual. *Be sure to unplug and remove any S-100 boards during these steps.*

•Step 1. Build the custom memory management control circuit as shown in figure 3 on a small vector board. Set this board to one side. It will be used in a later step.

•Step 2. (This step moves the display memory data output signals from the S-100 bus to the Sol's internal bus.) Lift all the output pins of the tri-state I.C.s (see drawing 4) U29 and U89 and tie to the internal bus signal INT 0—INT 7, (see drawing 1). The internal bus runs all over the Sol mother board; use any handy INT 0—INT 7 signals to connect to. Be sure to mark-up changes and keep a accurate set of prints of your computing system.

PIN 13 of U89 (74LS367) to "INT 0" PIN 10 of U79 (74LS253)
PIN 11 of U89 (74LS367) to "INT 1" PIN 6 of U79 (74LS253)
PIN 13 of U29 (74LS367) to "INT 2" PIN 10 of U65 (74LS253)
PIN 9 of U29 (74LS367) to "INT 3" PIN 6 of U65 (74LS253)
PIN 7 of U29 (74LS367) to "INT 4" PIN 10 of U78 (74LS253)
PIN 11 of U29 (74LS367) to "INT 5" PIN 6 of U78 (74LS253)
PIN 3 of U29 (74LS367) to "INT 6" PIN 10 of U66 (74LS253)
PIN 5 of U29 (74LS367) to "INT 7" PIN 6 of U66 (74LS253)

•Step 3. (This step modifies the control of the internal/external multiplexer (U66, U78, U65, U79—see drawing 1) to allow the data from the display to get to the processor.)

Lift PIN 2 of U44 (74LS00) and tie it to PIN 1 of U44.

•Step 4. (This step moves the MWRITE signal of the internal RAM, so that it may be controlled by the memory management circuit.)

Lift PIN 9 of U44 (74LS00) and tie to PIN 14 of U46 (8T30). See drawing 4.

Lift PIN 13 of U24 (74LS04) and tie to PIN 14 of U46 (8T30). See drawing 2.

•Step 5. At this point the Sol computer should operate normally. Plug it in and try some programs that use the display; TARGET is a good test program. If the system does not work there is a wiring error, so double check everything and try again.

•Step 6. In step 8 the connection of the memory management circuit board is installed. Mount the memory management board and connect the circuit to VCC (+5 Vdc) and ground. This power comes from the Sol mother board. Connect to the following signals to the Sol. See figure 2 and drawing 2.

74LS74 PIN 1 (reset) to (S-100 signal POC) PIN 12 of U77.
74LS74 PIN 3 (clock) to (OUTPUT FCH) PIN 11 of U35.
74LS74 PIN 2 (data) to (S-100 signal D0) PIN 2 of U80.
7406 PIN 8 to PIN 3 of U34.
7406 PIN 1 and PIN 13 to PIN 6 of U23.
7406 PIN 2 to (S-100 signal MWRITE) PIN 11 of U50.
7406 PIN 12 to (S-100 signal FRDY) PIN 1 of U49.

This completes the modification of the Sol.

*Step 7. The system should operate normally; retest as in step 5. If there are any problems check the memory management flip-flop to make sure the Power On Clear (POC) resets it to a low level at PIN 5.

Theory of Operation

When the computer is first turned on the memory control flip-flop is cleared via the Power-On-Clear signal (POC). This signal is also generated when a restart is performed (holding both the upper case and repeat keys down). The Sol will operate normally with the Solos/display RAM/ROM memory block enabled.

The memory control flip-flop controls accesses (reads/writes) to the C000—CFFF hex memory block. This block 4K Solos/display RAM/ROM or a 4K RAM (can be part of a larger memory plane) memory on the S-100 bus. In other words the memory control flip-flop switches in the internal Solos memory or the external S-100 memory.

Operation

Software control of the memory management flip-flop is accomplished via the output instruction OUT FC and bit 0. If bit 0 is set to a zero (0) then this is normal Sol operation. If bit 1 is set to a one this enables the memory on the S-100 bus.

The programming example illustrates how to have a full 65K RAM system and use the Solos utilities with CP/M . The cold boot switches off the internal memory and turns on all RAM external memory.

Software Rules

These rules should be kept in mind when using this system.

1) Do not switch to the internal memory (Solos) if the STACK is in the C000—CFFF address area. Save the stack first, or the program will not be able to find its way back.

2) Do not switch to the internal memory from insided the C000—CFFF address area. ■

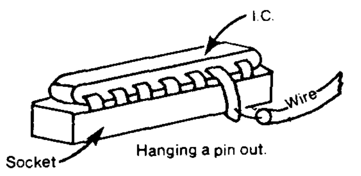


Figure 1.

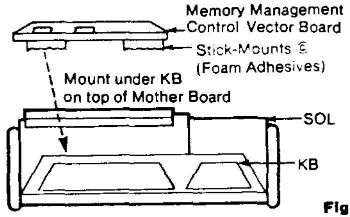


Figure 2.

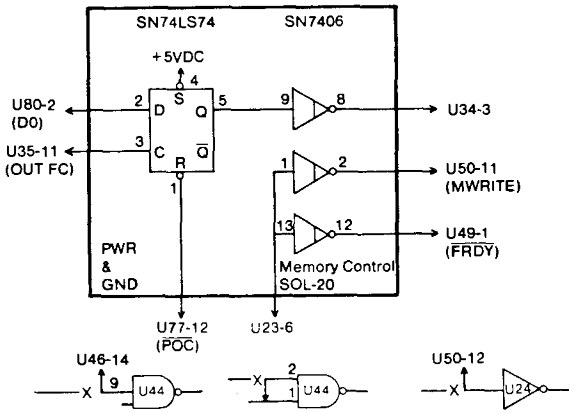


Figure 3.

CON'T ON PAGE 4

The following is an example of a CP/M interface using the memory management modification.

```
;
memsw   equ     0FCH                ;output that control the memory
solon   equ     0                   ;solos on, normal operation
solof   equ     not solon and 0FFH      ;enables all ram
;
sinp    equ     0C01FH              ;SOLOS entry points
sout    equ     0C019H
aout    equ     0C01CH
ainp    equ     0C022H
stat    equ     0FAH                ;Keyboard status
char    equ     0FCH                ;Keyboard data in
;
boot:   lxi     sp,stack

        mvi     a,solof             ;turns off solos rom/ram area
        out     memsw
          .
          .
          .

        jmp     cpm;
;
wboot:  lxi     sp,stack

        mvi     a,solof             ;turns off solos
        out     memsw
          .
          .
          .
        jmp     gocpm
;
;    I/O ROUTINES
;=========================================================
;
const:  in      stat                ;Sol KB
        cma
        ani     1
        rz
rtn:    mvi     a,0ffh
        ret
;
conin:  in      stat                ;Sol KB
        ani     1
        jnz     conin
        in      char
        ora     a
        jz      conin
        ret
;
reader: mov     a,1                 ;serial port
        lxi     h,ainp
        jmp     memctrl             ;send it to solos
;
punch:  mov      b,c                 ;for adout
        mvi      a,1                 ;serial port
        lxi      h,aout
        jmp      memctrl ;send it to solos

;       MEMCTRL
;       memory control
;       26 nov  79
; This program maps the solos area (C000 - CFFF) on & off.
; To allow for 64k byte operation of the Sol and still have
; access to the Solos software contain in the C000-CFFF area
; which includes the VDM, the following procedure is required.
;
; Power On Clear enables the Solos area so the first thing
; the boot program should do is turn off the Solos area.
;
; NOTE: THIS PROCEDURE MUST EXECUTE OUTSIDE OF THE C000-CFFF
;       ADDRESS SPACE.
;
; Program calling sequence example:
;**************************************************************
conout  push    h                   ;save h we need it
        lxi     h,sout              ;vector to vdm in solos
        call    memctrl
        pop     h                   ;restore h
        ret
;**************************************************************
;
memsw   equ     0FCH                ;output that control the memory
solon   equ     0                   ;solos on, normal operation
solof   equ     not solon and 0FFH      ;enables all ram
;
memctrl shld    vector+1            ;store vector
        lxi     h,0
        dad     sp                  ;get stack
        shld    stkreg              ;save it
        lxi     sp,stack            ;get a local stack outside
                                    ;C000 - CFFF range
        mvi     a,solon             ;turn on solos so we have access
        out     memsw
vector  call    0                   ;get set on entry
        mvi     a,solof             ;turn off solos area
        out     memsw
```

```
        lhld    stkreg              ;recover org stack
        sphl
        ret
stkreg  dw      0
        ds      20
stack   dw      0

        end
```

HARDWARE INTERRUPTS FOR THE SOL

by Stan Sokolow

Leon Winter in Bukidnon, Philippines, has written asking about using interrupts to drive his Hytype printer off of a parallel port. Other Proteus members have from time to time asked about implementing interrupts on the Sol. We haven't yet had anyone actually write a how-to article on adding vectored interrupts to the Sol, but I would sure like to have one.

In this issue, Dr. Howe has a letter mentioning he has developed an interrupt driven real-time multi-tasking system with his Sol/Helios. I've asked him to send me details so I can write an article on it. If anyone else has done something with interrupts, please let me know.

Just to get people thinking about interrupts, here's a little background information.

The way the Sol and most of today's microcomputers handle input/output is known as the polling method. When the program wants to read the keyboard, it first issues an input instruction directed at the keyboard's status port address. This puts the keyboard status byte into a register inside the 8080 microprocessor, where the program can test it. One bit in the byte is assigned to the flag which signals if a key has been pressed since the last time the keyboard data was read. The program keeps looping through the status read and test until the flag indicates a key has been pressed. Then the keyboard data port is read, obtaining the ASCII code for the key, and the program moves on.

The trouble with this method is that the program can do very little while it is in the loop, unless you are willing to risk loss of a keystroke by not reading the keyboard soon enough before another key is pressed.

Wouldn't it be nice if the processor could set things up so that the keyboard taps the processor on the shoulder, so to speak, when it needs to be read. The processor could go on doing other more useful work between keystrokes, without risking loss of data. This method is known as the interrupt method, and the "tap on the shoulder" is the interrupt signal.

The 8080 processor has a pin which can be used to signal that the current program is to be interrupted while a special subroutine is called. The programmer will set up the subroutine (known as the interrupt service routine) to do the desired task (read the keyboard in our example) and then return to the interrupted main program just where it left off.

Actually, the 8080 is able to have many interrupt service routines. The hardware that initiates the interrupt signal can supply the 8080 with either a RESTART instruction or a CALL instruction to designate where the interrupt service routine is located. This is known as a vectored interrupt, since the hardware can point (vector) the processor to the right address.

The Sol has no device for placing the RESTART or CALL onto the data bus, so it can only do polled input/output. However,

CON'T ON PAGE 23

PerSci 277 to 270 Conversion

by David Reis

March 25, 1982

It may be of interest to PROTEUS NEWS subscribers that PerSci 277 Disk Drives are being sold at very reasonable cost by Qantel Corp. (Hayward, CA). Actually, the drives are being sold by bid, but due to the lack of the market Qantel had anticipated for the drives they are accepting the minimum bid of $200.00. These drives are being sold "as is" and virtually all of them need service; however, without guaranteeing it Qantel states that most drives only need to be aligned. I've bought three of these drives and essentially found this to be the case.

There are five items that someone who purchases one of these drives from Qantel should be aware of if they intend to use it in a Helios system:

First: these drives have duel density data separators installed and the Helios requires the single density data separator. This is the small PCB on the left side of the drive towards the top front corner. The single density separator is necessary because the duel density separator decodes the READ DATA expecting Modified Frequency Modulation (MFM) formated data while single density (eg. Helios) is recorded using "simple" Frequency Modulation (FM) format. These two formats are not at all compatable. A new single density data separator from PerSci costs about $100; used ones can be bought for considerably less. Shops that service PerSci drives are likely sources of used data separators.

Second: there are four (4) signals that Qantel cut the traces for on the Data and Interface (D&I) PCB (the large PCB on the right hand side of the drive) between the 50 pin connector and the main body of the board:

PIN	SIGNAL
8	Seperated Index
16	Direct Head Load
20	Seperated Sector
46	Read Data

Qantel soldered leads from their own plugs to the cut traces (from the edge connector) which should be removed and then jumpers added to restore continuity to the 50 pin connector. This requires removing the two hold-down screws at the bottom corners of the D&I PCB and swinging the board up on the hinges at its top to get at the solder side of the board.

Third : the front bezel diskette eject switches are not active as Qantel relied on remote eject signals from the 50 pin connector (from the plugs mentioned in the above paragraph). The following wiring changes are one method to activate these switches:

Relay 0 / Switch 0 (Left bezel switch, forward relay*)

1) Remove the large white wire jumper between the relay contact with the small white wire and the relay reed.
2) Move the violet wire from the contact with the gray wire to the contact with the white wire.
3) Attach the green wire** from switch 0 to the relay contact with the orange wire.

Relay 1 / Switch 1 (Right bezel switch, rear relay*)

1) As above.
2) As above.
3) Attach the brown wire** from switch 1 to the relay contact with the brown wire.

* Both relays are found screwed to the bottom plate at the rear of the drive.
** These wires (from the bezel switches) are attached to the third plug Qantel added to the drive.

Fourth: the write protect circuit is active and depending on personal preference may or may not be defeated. To defeat this feature simply unplug P19 & P20 (Write Protect 1 & 0) on the top row of connectors of the Data and Interface PCB.

Fifth: Qantel/PerSci 277 disk drive D&I PCBs are configured for double density using soft sectoring (D&I PCB 200263-012). Processor Technology/PerSci 270/277 disk drive D&I PCBs were configured for single density using hard sectoring (D&I PCB 200263-007). The following changes/deletions should be made to configure these drives for use in a Helios system:

PerSci/Qantel 277 ----> PerSci/Processor Technology 270
(D&I PCB 200263-012) (D&I PCB 200263-007)

1) Change R88 & R94 to 4.75 Kohms (1/4 W, 5%).
2) Change C40 & C41 to 1 microfarad (Tantalum, 35V) (All of the above components are found just to the right and left of U10 and establish the index & sector pulse widths from it).
3) Change U11 (select Module DIP) to one described in the Helios II Disk Memory System Manual, Section 8, Fig 15; or, if only one duel drive is in the system simply Delete U11.
4) Jumpers:

	Delete	Add	270 Function
a)	W-X	W-Y	Sector Sep One Shot Trig 0
b)	AB-AC	AB-AA	Sep. Sector Pulse Output
c)	AD-AE	AD-AF	Sector Sep One Shot Trig 1
d)	AH-AJ	AH-AK	Sep. Index Pulse Output
e)	AW-AV	AW-AU	+5 to U11,14
f)	BH-U11,10	BH-BJ	Ties Sep Sector to Data Sep
g)	U32,5-U32,1	-----	

5) Sector 0: Jumper 32 to the pad immediatly to its right (to the right of U27).
Sector 1: Jumper 32 to the pad immediatly to its right (to the left of U16).

Additionally, the following "directory" of sources for various Helios components may be of interest to those who would like to build a Helios system.

PerSci 277 Duel Disk Drive (as is). . . . . ca. $200.00
Richard Abbas, Purchasing
Qantel Corp. (415) 887-7777 ex 2318
Hayward, CA

Helios Power Supply PCB (bare board). $10.00
David Tanconie
Silicon Technology (415) 967-8250
Box 70421
Sunnyvale, CA 94086

Helios (and Sol) Transformers $15.00
Dealin' Electronics
735 Loma Verde (415) 493-5930
Palo Alto, CA 94303

CON'T ON PAGE 6

IMSAI Cabinet for one PerSci Disk Drive $50.00
    WW Component Supply Inc.
    1771 Junction Ave.   (408) 295-7171
    San Jose, CA 95112
    (The Helios Power Supply PCB & Transformer fit in this cabinet very conveniently. The Helios Indicator PCB is easily adapted to this cabinet by drilling some holes in the front panel for the LEDs and attaching a photo-reproduction of the indicator panel depicted on pg. 4-4 of the Helios II User's Manual for front panel dress).

Assorted Helios Components and Systems
    Charles Terry
    Special Pals   (713) 666-1316 Home
    4211 Bellaire Blvd.   (713) 666-3515 Work
    Huston TX, 77025
    (Such things as Helios controllers & Formatter boards, indicator PCBs, and some cabinet parts).

Helios Documentation & Software
    PROTEUS

This information is only provided for reference and its accuracy can not be guaranteed.

David Reis

David Reis
1843 Vassar Avenue
Mountain View, CA 94043

LOCATING SAA 1027 CHIP AND CONVERTING ECBASIC TAPE TO DISK
by John Whiting

For those who are interested in such things (assuming this letter gets published), my system is a Rev. E SOL with Measurement & Systems Controls 65 K memory board, Objective Design's Programmable Character Generator, Ackerman Digital's Noisemaker board, Microcomplex's North Star look-alike disk controller, a Paper Tiger 560 G, and one and a half BASF disk drives that I bought used.

I say "one and a half drives" because the motor controller chip has died on one of them. It's a SAA 1027 chip, whatever the devil **that** is. I've got BASF trying to locate one for me, but it couldn't hurt to ask the readers of PROTEUS to help, and it might help someone else who is trying to find a source for this (and other obscure) part(s).

The other thing on my mind is that I keep hearing about a program that is supposed to convert ECBASIC to a CP/M-compatible BASIC, and include a utility for converting ECBASIC tape files to disk files. While I'm using McDos (Microcomplex's N* look-alike) right now, I'd like to go to CP/M eventually, and I think I'd like a disk version of ECBASIC better than I would some of the others available. (If anyone is interested, I'm trying to write a machine language program that will fool N* BASIC into thinking that input from an ECBASIC text-mode tape is coming from the keyboard. This would allow fairly easy conversion from tape to disk without having to worry about single-byte keyword conversions.)

Guess that's about it. Keep up the good work!

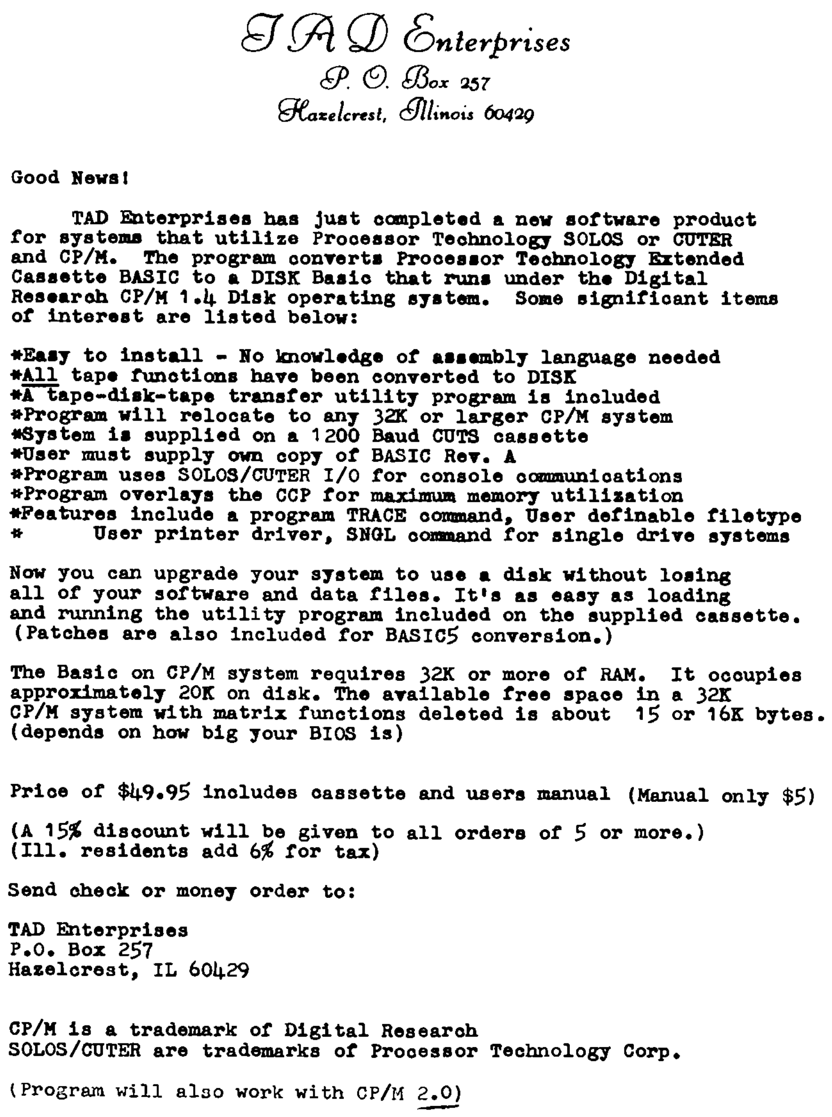
Regards,

John A. Whiting

John A. Whiting
230 S. Coronado St. #21
Los Angeles, California
3/4/82   90057

CONTINUED....

(Ed. note: Funny you should ask about the ECBASIC conversion to CP/M. The following flyer describes the product. Many Proteus members have written in praise of it. It is still available from TAD Enterprises. An improved, disk-oriented version of this BASIC adapted to CP/M and will be available in the future.--Stan.)



TAD Enterprises
P. O. Box 257
Hazelcrest, Illinois 60429

Good News!

TAD Enterprises has just completed a new software product for systems that utilize Processor Technology SOLOS or CUTER and CP/M. The program converts Processor Technology Extended Cassette BASIC to a DISK Basic that runs under the Digital Research CP/M 1.4 Disk operating system. Some significant items of interest are listed below:

***Easy to install - No knowledge of assembly language needed**
***<u>All</u> tape functions have been converted to DISK**
***A tape-disk-tape transfer utility program is included**
***Program will relocate to any 32K or larger CP/M system**
***System is supplied on a 1200 Baud CUTS cassette**
***User must supply own copy of BASIC Rev. A**
***Program uses SOLOS/CUTER I/O for console communications**
***Program overlays the CCP for maximum memory utilization**
***Features include a program TRACE command, User definable filetype**
***   User printer driver, SNGL command for single drive systems**

Now you can upgrade your system to use a disk without losing all of your software and data files. It's as easy as loading and running the utility program included on the supplied cassette. (Patches are also included for BASIC5 conversion.)

The Basic on CP/M system requires 32K or more of RAM. It occupies approximately 20K on disk. The available free space in a 32K CP/M system with matrix functions deleted is about 15 or 16K bytes. (depends on how big your BIOS is)

Price of $49.95 includes cassette and users manual (Manual only $5)

(A 15% discount will be given to all orders of 5 or more.)
(Ill. residents add 6% for tax)

Send check or money order to:

TAD Enterprises
P.O. Box 257
Hazelcrest, IL 60429

CP/M is a trademark of Digital Research
SOLOS/CUTER are trademarks of Processor Technology Corp.

(Program will also work with CP/M <u>2.0</u>)

AN ADS NOISEMAKER PROGRAM FOR A 38-KEY PIANO

by Frank Gizinski

As far as my Sol 20 goes I'm sort of a Rip Van Winkle. Shortly after finishing the kit I put a hole in my car windshield with the top of my head in an accident. The Sol was all right but I had memory damage. About the time I was coming around again I saw Mr. Hallen's article in Kilobaud where he mentioned moving his sense switches to the front panel. Sounded like a great idea at the time, but, after a few hours of spastic desoldering I had fairly well ruined the Sol's main circuit board. This was about the time Processor Technology went out of business so I got a replacement board from Advanced Computer Products, which didn't work. I then handed it to what was then known as the Milwaukee Computer Store. I got it back nine months later and it still didn't work but, eventually, I found and repaired an open connection between U47-5 and U45-10 and I was back in business, complete with sense switches on the front panel where the Sol logo used to be.

In the meantime I saw the article in the December, 1979, Microcomputing "Extending the Altair Bus" so I got myself a big blue Vector UP1 box, a Sunny International power supply and an 88 slot motherboard, on sale somewhere. The board had the name Godbout on it and seems to have been made for the Vector box. Then I soldered four, five foot 50 conductor cables to two cutdown blank S-100 boards, every other conductor being a ground but no power connections made. That was something I wouldn't care to try again, it was like giving mouth-to-mouth resuscitation to a boa constrictor. The whole thing turned out pretty well, I was lucky with my hacksaw work and the thing even looks pretty good. There are cutouts on one side of the box for ten RS-232 connectors, on the other side I found room for a plug, an on-off switch, a five inch speaker with a volume control, a Sonalert, and a surplus Sol fan, although my two Artec 32k boards aren't cooking each other in there any more with all that room. I call the thing Gizinski's Pole-Lution, no rights reserved.

I'm still running cassette and have two little control boxes with Read-Write and Manual-Computer switches. I also have 2" speakers, with on-off switches in the boxes. The speakers come in handy sometimes to let you know that you remembered to switch to Read when you're saving something and such. I note that some people have mentioned that the GRT Microsoft BASIC tape I/O didn't work. I finally, by accident, read something back in that I had saved using GRT one day and eventually found on page 11 of their manual that you had to make sure "the leader tone is present before inputting from the tape.", soyou need a speaker to hear it. Can't see any advantage to the GRT BASIC though. The only problem I ever had with P T BASIC was with string operators until I figured out what Mr. Moseley was trying to tell me in his "Programming Quickies" on strings in Proteus, Vol. 2, #5.

Yours truly,
Frank Gizinski

The following is a program for the ADS Noisemaker which will give you a 38 key piano. I've got my Noisemaker addressed at I/O 0F4H,the keys used are "Z" through "/" and "Q" through "Return". Q is middle C and "S" "D" "G" "H" "K" "L" ";" "2" "3" "5" "7" "6" "9" "0" " " and "[" are the sharps. I calculated the frequency for the notes assuming that the clock frequency for the Sol is 2.045MHZ, sound about right but I can't find the darned tuning fork that's around here someplace. I punched out dots from a white self-adhesive label and put them on the front of the sharp keys to remember where they are. ..

ADS NOISEMAKER PROGRAM FOR A 38-KEY PIANO

```
7000                0010          ORG     7000H
7000                0020 *
7000                0030 *     NOISEMAKER PIANO
7000                0040 *     11/02/81
7000                0050 *     Frank Gizinski
7000                0060 *
7000                0070 NOISE    EQU     0F4H    S1 IS SET FOR I/O 0F4H
7000 2A CC 70       0080 A1       LHLD    VALAD
7003 EB             0090          XCHG
7004 2A CA 70       0100          LHLD    CHARD
7007 06 26          0110          MVI     B,38    NUMBER OF NOTES
7009 DB FA          0120 IPT      IN      0FAH    SAME AS SINP
700B 2F             0130          CMA
700C E6 01          0140          ANI     01
700E CA 09 70       0150          JZ      IPT
7011 DB FC          0160          IN      0FCH
7013 FE 1B          0170          CPI     1BH     ESC TO MONITOR
7015 CA 04 C0       0180          JZ      0C004H
7018 BE             0190 FIND     CMP     M
7019 CA 26 70       0200          JZ      SOUND
701C 23             0210          INX     H
701D 13             0220          INX     D       LINE UP BOTH COARSE AND FINE
701E 13             0230          INX     D       VALUES WITH CHAR
701F 05             0240          DCR     B
7020 CA 00 70       0250          JZ      A1
7023 C3 18 70       0260          JMP     FIND
7026 3E 07          0270 SOUND    MVI     A,07    ENABLE CONTROL
7028 D3 F4          0280          OUT     NOISE
702A 3E FE          0290          MVI     A,0FEH  ENABLES TONE A, REGISTER IS
702C D3 F5          0300          OUT     NOISE+1 NEGATIVE TRUE
702E 3E 08          0310          MVI     A,08    AMPLITUDE A REGISTER
7030 D3 F4          0320          OUT     NOISE
7032 3E 10          0330          MVI     A,10H     SET TO VARIABLE, DETERMINED
7034 D3 F5          0340          OUT     NOISE+1   BY ENVELOPE GENERATOR
7036 3E 00          0350          MVI     A,00    CHANNEL A TONE
7038 D3 F4          0360          OUT     NOISE
703A 1A             0370          LDAX    D       FINE TONE
703B D3 F5          0380          OUT     NOISE+1
703D 3E 01          0390          MVI     A,01    CHANNEL A TONE
703F D3 F4          0400          OUT     NOISE
7041 13             0410          INX     D
7042 1A             0420          LDAX    D       COARSE TONE
7043 D3 F5          0430          OUT     NOISE+1
7045 3E 0C          0440          MVI     A,0CH   ENVELOPE PERIOD CONTROL
7047 D3 F4          0450          OUT     NOISE
7049 3E 17          0460          MVI     A,17H     SET NOTE LENGTH, VARY IT AND
704B D3 F5          0470          OUT     NOISE+1   SEE WHAT SOUNDS RIGHT TO YOU
704D 3E 0D          0480          MVI     A,0DH   ENVELOPE SHAPE CONTROL
704F D3 F4          0490          OUT     NOISE
7051 3E 00          0500          MVI     A,00      I THINK THIS IS THE CLOSEST
7053 D3 F5          0510          OUT     NOISE+1   TO A PIANO SHAPE
7055 C3 00 70       0520          JMP     A1      BACK FOR NEXT KEY PLAYED
7058 5A             0530 CHAR     DB      5AH
7059 53             0540          DB      53H
705A 58             0550          DB      58H
705B 44             0560          DB      44H
705C 43             0570          DB      43H
705D 56             0580          DB      56H
705E 47             0590          DB      47H
705F 42             0600          DB      42H
7060 48             0610          DB      48H
7061 4E             0620          DB      4EH
7062 4D             0630          DB      4DH
7063 4B             0640          DB      4BH
7064 2C             0650          DB      2CH
7065 4C             0660          DB      4CH
7066 2E             0670          DB      2EH
7067 3B             0680          DB      3BH
7068 2F             0690          DB      2FH
7069 51             0700          DB      51H     Q, MIDDLE C
```

CON'T ON PAGE 8

```
706A 32          0710        DB     32H
706B 57          0720        DB     57H
706C 33          0730        DB     33H
706D 45          0740        DB     45H
706E 52          0750        DB     52H
706F 35          0760        DB     35H
7070 54          0770        DB     54H
7071 36          0780        DB     36H
7072 59          0790        DB     59H
7073 37          0800        DB     37H
7074 55          0810        DB     55H
7075 49          0820        DB     49H
7076 39          0830        DB     39H
7077 4F          0840        DB     4FH
7078 30          0850        DB     30H
7079 50          0860        DB     50H
707A 40          0870        DB     40H
707B 5E          0880        DB     5EH
707C 0D          0890        DB     0DH
707D 5B          0900        DB     5BH
707E 18          0910 VALUE  DB     18H
707F 05          0920        DB     05
7080 CF          0930        DB     0CFH
7081 04          0940        DB     04
7082 8A          0950        DB     8AH
7083 04          0960        DB     04
7084 49          0970        DB     49H
7085 04          0980        DB     04
7086 0B          0990        DB     0BH
7087 04          1000        DB     04
7088 D1          1010        DB     0D1H
7089 03          1020        DB     03
708A 9A          1030        DB     9AH
708B 03          1040        DB     03
708C 66          1050        DB     66H
708D 03          1060        DB     03
708E 36          1070        DB     36H
708F 03          1080        DB     03
7090 08          1090        DB     08
7091 03          1100        DB     03
7092 DC          1110        DB     0DCH
7093 02          1120        DB     02
7094 B3          1130        DB     0B3H
7095 02          1140        DB     02
7096 8C          1150        DB     8CH
7097 02          1160        DB     02
7098 68          1170        DB     68H
7099 02          1180        DB     02
709A 45          1190        DB     45H
709B 02          1200        DB     02
709C 24          1210        DB     24H
709D 02          1220        DB     02
709E 06          1230        DB     06
709F 02          1240        DB     02
70A0 E9          1250        DB     0E9H   MIDDLE C
70A1 01          1260        DB     01
70A2 CD          1270        DB     0CDH
70A3 01          1280        DB     01
70A4 B3          1290        DB     0B3H
70A5 01          1300        DB     01
70A6 9B          1310        DB     9BH
70A7 01          1320        DB     01
70A8 84          1330        DB     84H
70A9 01          1340        DB     01
70AA 6E          1350        DB     6EH
70AB 01          1360        DB     01
70AC 59          1370        DB     59H
70AD 01          1380        DB     01
70AE 46          1390        DB     46H
70AF 01          1400        DB     01
70B0 34          1410        DB     34H
70B1 01          1420        DB     01
70B2 22          1430        DB     22H
70B3 01          1440        DB     01
70B4 12          1450        DB     12H
70B5 01          1460        DB     01
70B6 03          1470        DB     03
70B7 01          1480        DB     01
70B8 F4          1490        DB     0F4H
70B9 00          1500        DB     00
70BA E7          1510        DB     0E7H
70BB 00          1520        DB     00
70BC DA          1530        DB     0DAH
70BD 00          1540        DB     00
70BE CD          1550        DB     0CDH
70BF 00          1560        DB     00
70C0 C2          1570        DB     0C2H
70C1 00          1580        DB     00
70C2 B7          1590        DB     0B7H
70C3 00          1600        DB     00
70C4 AD          1610        DB     0ADH
70C5 00          1620        DB     00
70C6 A3          1630        DB     0A3H
70C7 00          1640        DB     00
70C8 9A          1650        DB     9AH
70C9 00          1660        DB     00
70CA 58 70       1670 CHARD  DW     CHAR
70CC 7E 70       1680 VALAD  DW     VALUE
```

Frank Gizinski

Please make note of the correct phone number for Data Delay Devices Co. mentioned in the article "16 KRA Data Delay", Vol 4 #5/6, page 27----(201) 772-1106.

ENCYCLOPEDIA PROCESSOR TECHNICA IS READY

The Complete Guide to Processor Tech Products

We have just put finishing touches onto eleven volumes of the Encyclopedia we mentioned in prior issues, and we are planning two more. The full set of volume tables of contents and prices appear in the Proteus catalog inserted in the center of this issue.

You will notice that the EPT, as we call it, includes nearly all of the manuals and updates, along with items from Proteus News that are of general usefulness. There is some original material that has never been published before.

The service manuals will be especially helpful to those who do their own maintenance. The diagnostic programs mentioned in the manuals (SOLT and DISKT) are available from Proteus to complement the Encyclopedia.

We've worked hard to organize this mass of material and bind it in attractive, loose-leaf folders. As new material becomes available, we will print supplements to be inserted into the volumes. We are sure you will find this an invaluable resource in the years to come.

FIXING SOL TO "VANISH"

by Wayne Wilson

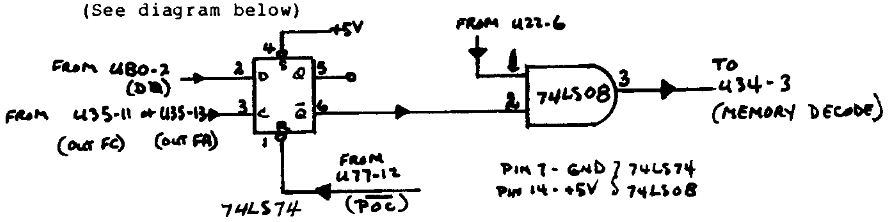
8022 - 117th Street
Delta, B.C. V4C 6A9
CANADA

Dear Stan,

I have been running a 56K CP/M system with Solos at F000H for a couple of years now and have often thought it would be nice to squeeze alittle more memory into my system. I run Morrow's DJ2D controller which resides at E000H, this of course has limited me to 56K. Your article in the May-August issue - 'Make Sol Vanish' seemed the answer. Burn new proms at F800H for the controller board, put Solos back at C000H, add another 6K of RAM for a total of 62K....but wait a minute...How the heck did you get the mod with the 74LS74 as shown to work. In my Sol, the connection from pin 6 of the LS74 (Q-NOT) to pin 3 of U-34 (Memory decode) seemed to be a conflict situation. In fact, it decoded C000, D000 and E000 the same (that is as C000). After scratching my head for awhile and looking at the schematic it seemed the only way to make this mod work was to gate the signal from pin 6&8 of U-22 to pin 3 of U-34. This theory was tested and it works fine. The chip used is a 74LS08 (or 7408) 'AND' gate. It also was piggy backed on another chip to pick up +5 (pin 14) and GND (pin 7). The gating was added as follows:

1. Remove U-34 and bend out pin 3 and carefully tin with solder.
   Re-insert U-34 in socket.
2. Connect a wire from pin 6 of U-22 to pin 1 of the 74LS08.
3. Connect a wire from pin 6 of the 74LS74 to pin 2 of the 74LS08.
4. Connect a wire from pin 3 of 74LS08 to pin 3 of U-34.

On POWER UP or RESET pin 2 of 74LS08 goes HIGH due to the LS74 F-F being reset. This allows the signal from 6&8 of U-22 to be gated thru to U-34 and be decoded. When Bit 0 of port FA is set to 1 and toggled into the flip flop, pin 2 of the 74LS08 goes LOW. This inhibits the gate and the O/P (pin 3 of 74LS08) also goes low. This prevents memory decode and presto Solos is gone. (See diagram below)



IMPROVED ASL2 MULTIWRITER DRIVER

by Wayne Wilson

In the May-August issue you printed my Custom I/O Printer Driver for an ASL2 Multiwriter with a Diablo Hytype 1 printer assembly. That version featured 'handshaking' and collected 'white spaces' which allows the print carriage to move horizontally to next print location in one big tab rather than by printing individual spaces. I have now updated this driver to also collect 'linefeeds' and move the paper vertically in one big tab rather than by individual linefeeds. The updated driver follows:

```
******************************************************************
*                           ASL2                                 *
* Custom I/O printer driver for Multiwriter printer with Hytypel*
* assembly.  Handshake for 1200 baud and collected spaces.       *
*                         (10 cpi)                               *
******************************************************************
coull   XRA     A
        STA     TAB
SEROUT  PUSH    H       ;H/L WILL BE USED
        PUSH    D       ;ALSO D/E
        LHLD    FFEED   ;GET THE CURRENT COUNT
        XCHG            ;FOR L/FEEDS AND FOR
        LHLD    RIGHT   ;SPACES
        MOV     A,C
        CPI     20H     ;CHECK FOR A SPACE
        JZ      INC10   ;HOLD CARRIAGE, COLLECT SPACES
        LDA     SPACES  ;CHECK TO SEE IF
                        ;LAST PRINTABLE CHARACTER
        CPI     01      ;WAS A SPACE.
        JNZ     LFEED   ;IF NOT GO CHECK FOR LINE FEED
                        ;
                        ;AFTER COLLECTING SPACES
                        ;SEND OUT ONE BIG TAB.
                        ;
        MOV     A,C
        STA     TEMPRY  ;SAVE THE LAST PRINTABLE
                        ;CHARACTER IN AFTER SPACES
        CALL    BFLAG   ;SEND OUT THE BUFFER FLAG
        CALL    WAIT    ;LOOK FOR ANSWER-BAK, THEN MOVE
        MVI     C,01CH  ;LEAD IN CODE FOR A MOVE COMMAND
        CALL    SOUT
        MOV     C,H     ;NOW SEND OUT DIRECTION
        CALL    SOUT    ;    OF MOTION
        MOV     C,L     ;AND HOW FAR TO MOVE THE
        CALL    SOUT    ;    CARRIAGE
        XRA     A       ;TIME TO RESET
        STA     RIGHT   ;THE TAB REGISTER
        STA     RIGHT+1
        STA     SPACES  ;AND LOWER THE FLAG
        LDA     TEMPRY  ;GET CHARACTER BEING STORED
        MOV     C,A     ;PUT IT IN C REG.
        JMP     LFEED   ;CHECK IF CHAR. A L/F
SOUT    CALL    colpt   ;SEND A CHARACTER OUT
        LDA     CHRCNT  ;GET CHARACTER COUNT
        INR     A       ;BUMP THE COUNT
        STA     CHRCNT  ;STORE NEW COUNT
        CPI     60      ;SENT 60 CHARACTERS?
        JZ      WAIT    ;IF SO STAND-BY
        CPI     55      ;IF 60, IS IT 55?
        JNZ     BACK    ;IF NOT, GET MORE
BFLAG   MVI C,  05H     ;IF=55, SEND OUT BUFFER MARKER
        CALL    colpt   ;TO DIABLO
BACK    LDA     SPACES  ;GET FLAG
        CPI     1       ;CHECK FOR SPACES
        JZ      RTRN    ;GET MORE MOTION DATA
        LDA     LFD     ;GET L/F FLAG
        CPI     1       ;SEE IF IT IS RAISED
        JZ      RTRN    ;GET MORE VERT. MOTION DATA
POPS    POP     D
        POP     H
RTRN    RET             ;BACK FOR MORE
WAIT    IN      0F8H    ;GET STATUS
        ANI     040H    ;MASK FOR SOL SER. DATA READY
        JZ      WAIT    ;LOOP ROUND TILL READY
        IN      0F9H    ;GET DATA
        ANI     07FH
        CPI     07FH    ;IS IT DIABLO ANS-BAK
        JNZ     WAIT    ;IF NO TRY AGAIN
        XRA     A       ;AFTER ANS-BAK, RESET
        STA     CHRCNT  ;RESET THE COUNTER
        JMP     BACK    ;RETURN FOR MORE
```

CON'T ON PAGE 10

```
;
;SPACES COLLECTED HERE BASED ON 6Ø INCREMENTS PER
;INCH (5 increments per char. for 12 cpi)
;     (6 increments per char. for 1Ø cpi)
;
INC1Ø   MVI     A,1     ;RAISE THE `SPACES` FLAG
        STA     SPACES
        LDA     TAB     ;FIND OUT IF WE ARE
        CPI     1       ;PRINTING 12 cpi
        JZ      INC12   ;IF YES, SWITCH TO 12
        MOV     A,L     ;INCREMENT MOVE COUNT BY 6
        ADI     6       ;6/6ØTH INCREMENT FOR 1Ø CPI
        MOV     L,A
        JMP     MASK    ;CHECK THE HIGH ORDER BIT
INC12   MOV     A,L     ;INCREMENT MOVE COUNT BY 5
        ADI     5       ;5/6ØTH INCREMENT FOR 12 CPI
        MOV     L,A
MASK    ANI     8ØH     ;MASK FOR HIGH ORDER BIT
        CPI     8ØH     ;WHEN COUNT OVER 127
        JZ      HROUT   ;GO INCREMENT "H" REG.
        SHLD    RIGHT   ;IF NOT, STORE THE COUNT
        JMP     POPS    ;GO BACK FOR MORE STUFF
HROUT   MOV     A,L     ;IF COUNT OVER 127
        SUI     128     ;BY HOW MUCH
        MOV     L,A     ;PUT REMAINDER BACK IN L
        INR     H       ;COUNT EXTRA IN `H`
        SHLD    RIGHT
        JMP     POPS    ;THEN GO BACK FOR MORE
LFEED   MVI     A,ØAH
        CMP     C       ;COMPARE FOR LINEFEED
        JNZ     CTAB    ;CHECK IF LAST CHAR. WAS L/FEED
        MVI     A,1
        STA     LFD     ;RAISE THE L/FEED FLAG
        MOV     A,E     ;INCREMENT VERTICAL MOTION INDEX
        ADI     8       ;8 INCR./LINE - 6 LPI
        MOV     E,A
        ANI     8ØH     ;MASK FOR HIGH ORDER BIT
        CPI     8ØH     ;IF COUNT OVER 127
        JZ      DROUT   ;GO INCREMENT D REGISTER
        JMP     STOR    ;OTHERWISE STORE IT
DROUT   MOV     A,E     ;IF COUNT OVER 127
        SUI     128     ;BY HOW MUCH
        MOV     E,A     ;PUT REMAINDER BACK
        INR     D       ;EXTRA COUNTS IN D
STOR    XCHG            ;SWAP H/L AND D/E
        SHLD    FFEED   ;STORE THE COUNT
        XCHG            ;PUT EVERYTHING BACK
        JMP     POPS
CTAB    LDA     LFD     ;GET FLAG
        CPI     1       ;CHECK FOR PREVIOUS L/FEEDS
        JNZ     SOUT    ; IF NOT, SEND CHARACTER OUT
VTAB    MOV     A,C
        STA     VTEMP   ;SAVE CHARACTER
        CALL    BFLAG   ;SEND OUT BUFFER FLAG
        CALL    WAIT    ;WATCH FOR ANS-BAK, THEN GO
        MVI     C,Ø1CH  ;LEAD-IN CODE FOR MOTION
        CALL    SOUT
        MOV     C,D     ;SEND OUT DIRECTION
        CALL    SOUT    ;    OF  MOTION
        MOV     C,E     ;AND HOW FAR TO MOVE
        CALL    SOUT    ;    THE PAPER
        MVI     A,1ØH
        STA     FFEED+1 ;RESET PAPER MOTION BIT
        XRA     A
        STA     FFEED   ;RESET FORMFEED REGISTER
        STA     LFD     ;RESET FLAG
        CALL    DELAY   ;STALL FOR BIG TABS
LDA     VTEMP   ;GET CHARACTER BEING HELD
        MOV     C,A
        JMP     SOUT    ;SEND IT OUT
DELAY   LXI     H,3FFFH ;SET UP COUNTER FOR DELAY
        DCX     H       ;COUNT DOWN
        MOV     A,H     ;CHECK FOR ZERO
        CPl     ØØH
        JNZ     DELAY   ;NOT ZERO, MORE DELAY
        RET
CHRCNT  DB      ØØH
SPACES  DB      ØØH     ;1 IF PREVIOUS CHARACTER WAS A SPACE
RIGHT   DW      ØØØØH   ;HERE WE COUNT ALL THE COLLECTED SPACES
TEMPRY  DB      ØØH
TAB     DB      ØØH     ;12 cpi IF FLAG IS '1'
FFEED   DW      1ØØØH   ;COUNT VERTICAL INCREMENTS AND SET MOTION
LFD     DB      ØØH     ;LINEFEED FLAG
VTEMP   DB      ØØH
;
*****************************************************************
*                                                               *
*    Set flag for printing at 12 cpi on ASL2 Multiwriter.       *
*                                                               *
*****************************************************************
coul2   MVI     A,1     ;Set up flag for
        STA     TAB     ;printing 12 cpi.
        JMP     SEROUT  ;To main print routine
*****************************************************************
*                                                               *
*             SEND CHARACTER OUT SERIAL PORT                    *
*                                                               *
*****************************************************************
colpt   IN      ØF8H            ;GET SOL SERIAL STATUS
        RAL                     ;PUT HIGH BIT IN CARRY
        JNC     colpt           ;LOOP UNTIL BUFFER EMPTY
        MOV     A,C             ;GET CHARACTER READY
        OUT     ØF9H            ;SEND IT OUT
        RET
```

REVIEW OF VIO-X VIDEO BOARD

by Wayne Wilson

For the information of other Sol users, I have been using a VIO-X 80x24 Intelligent Video Board for several months now. The board was easy to implement and ran first time in my SOL without a problem. The addition of the 8Øx24 display was like having a new computer. I personally feel that the 8Øx24 display is that extra something missing from the Sol.

The board is S-100, well layed out and nicely silk screened. It uses the Intel 8275 CRT controller with an on board 8085 microprocessor and 4K of RAM. The board operates outside the system via 2 ports, so does not use any system RAM. The display rate is very fast (effectively 8Ø,ØØØ baud according to the manual). At any rate it is fast and I didn't feel slowed down after the memory mapped display of the Sol.

The display is a 80x25 line format (25th line is a status line) using a 5x7 character set in a 7x1Ø dot matrix. There is a full upper and lower case ASCII alphanumeric character set (true descenders) plus 32 special characters for Escape and Control characters. An optional 2732 character generator is available with an alternate 7x1Ø contiguous line and block graphics set. (I have not tried this option).

A second board, the VIO-X2 is also offered with a 7x7 character set ib a 9x1Ø matrix allowing high-resolution characters. This version also includes expanded firmware for block mode editing and light pen location. Contiguous graphics characters are not supported. (I have not used this version). Just included for info purposes.

Both versions support a full set of control characters and escape sequences...including controls for video attributes, cursor XY Positioning, cursor toggle (ON/OFF) and scroll speed. There is

CON'T ON PAGE 11

an on board Real Time Clock which displays on the bottom right hand corner of the screen on the 25th line. It is a 24 hour clock and is easily set from the keyboard. (ie: ESC>112532 produces the following display 11:25:32). The clock can also be read from the host system. Video attributes are:

Flash Character
Inverse Character
Underline Character
Dim Character

The board can be addresses to any port pair in the IEEE-696 host system (S-100). Inputs are provided for parallel keyboard and light pen and an output for audio signalling.

Other features:

- High speed operation
- Port mapped
- Forward/Reverse scroll of 2 page screen
- Transparent mode
- Interrupt operation compatible with MP/M
- Blinking or Non-blinking cursor

The video attributes possible are as follows:

Normal
Dim
Blinking
Blinking dim
Inverse
Inverse dim
Inverse blinking
Inverse blinking dim
Underline
Underline dim
Underline blinking
Underline blinking dim
Underline inverse
Underline inverse dim
Underline inverse blinking
Underline inverse blinking dim

Basically the VIO-X emulates a Soroc terminal. I have run software written for a Soroc without any problem. The plus is that this board also has video attributes not available on the Soroc and some pretty flashy displays can be produced as a result for business applications. The 80x24 display is great for word processing applications as well.

The board comes assembled c/w a 32 page manual, parts list and schematic diagram. The manual is not too bad and no trouble was experienced in getting the board up and running with it, as well as trying out the various features.

I have tried other boards on the market prior to this and basically took a chance when ordering this board based on the description in the Advertisement. It sounded like it had the features I was looking for and the price was reasonable compared to others.

The board is supplied by W. W. Component Supply, Inc.
1771 Junction Ave.
San Jose, CA 95112
Tel:(408) 295-7171
Advertised price: $295.00 (U.S.).

CONTINUED....

I guess it's apparent I'm happy with this product. It is pleasing to get a product that lives up to the claims in the Ad. I am aware of several other people using this same video board also in Sols and to my knowledge they feel as I do, that it has added new life to the old beast.

Yours truly

Wayne Wilson

Electric Pencil Using N*DOS

by John Osudar

It's been too long since I last wrote you. You must be really busy, judging from the double issues of the newsletter. My SOL is sick, so I'm actually typing this on a real typewriter (!!!); so excuse please the blotches of Liquid Paper.
Several things I want to say, so here goes:
First, many thanks to Joe Maguire for his fine articles, especially on fixing the SOL.
Second, memory boards. My old 16KRA (original) recently breathed its last, after serving surprisingly well for 4½ years. In a desperate search for a replacement during the holidays, I came across a company called Memory Merchant, selling a 16K static for $174 including shipping. It took just over a week to get the board, and it sure seems to be worth the money. Interested readers should check Memory Merchant's ads in BYTE. I may even buy their 64K board, if I can scrape up the money! (By the way, if anyone is interested in an old non-functioning 16KRA, for spare parts, as a museum piece, or if they think they might be able to fix it, I'm quite willing to part with it!)
Third, regarding Electric Pencil I Version SS running with disk I/O. I've got Pencil relocated to an origin of 2D00H for my North Star system, and could supply instructions for relocating to any other page boundary (e.g. 100H for CP/M?) I also have a set of disk I/O routines that use the N* DOS to replace SOLOS cassette functions, allowing PENCIL to do disk I/O. If you will make my mailing address available in the newsletter, I'm willing to accept a limited amount of correspondence on this matter. By the way, a note to Bob Johnson of Boulder, CO (letter on p. 31 of V4#3/4): I get Pencil to send control characters to my Paper Tiger by using a little software printer driver, which transforms a two-character sequence (tilde followed by any character) to the corresponding control code. For example, to send control-P to enter proportional spacing mode, I include a tilde followed by P in the text. It's not perfect, but it works!
Finally, regarding Earl Dunham's letter about Bob Hogg, PTDOS, etc.: Why can't PROTEUS provide some financial backing for potential developers of new SOL hardware or software, or at least assure interested parties (such as Mr. Hogg) of some level of member interest in such products? We've all got enough invested in our SOL systems that improving our existing systems is preferable to scrapping them, and buying something new.
That's about it for this letter, Stan. Looking forward to the next newsletter, and continued prosperity for you and PROTEUS in 1982.

Yours truly,

P. O. Box 1451
Homewood, IL 60430-0451

John Osudar

24 X 80 CONVERSION AND THE NOVICE INSTALLER
by Philip N. Barnhart

Dear Stan:

I have had Micro Complex's 24 X 80 screen up and running for over a month now with my new BMC KG-12C monitor also sold by Micro Complex. I agree with all the favorable comment in the latest PROTEUS NEWS. It works great with my configuration which includes Discus 2D with CP/M 2.2.

I have one additional comment. I know one end of a soldering iron from the other by noting the temperature difference after it has been plugged in for a while. Yet I had no trouble following the installation instructions for the 24 X 80 board or the DPM. The whole thing was up and running in about 3 hours not counting the time it took to write the new interface I needed from CP/M (If your BIOS does not reference any of the internal SOLOS routines but only the jump table, the only modification of the BIOS which should be needed is to change the SOLOS reference from page C0 to F0).

Bob Hogg has been extremely helpful and available during the several times I called him, even at 9:30 pm when he helped me correct one minor installation problem over the phone.

The new screen almost doubles the amount of information which I can access at one time and makes the SOL much more useful for word processing and data entry than before. I use MINCE for program entry and word processing. Its split screen capabilities were lost on me with the old VDM. Half of the screen simply could not hold enough lines to be useful. Now half of the screen holds only one less line than the entire screen did before.

I can hardly wait for the Z-80 upgrade Bob Hogg is working on. It will allow us to run Z-80 software and at almost double the clock speed, or at the same slow clock speed if your memory is not up to it.

Philip N. Barnhart
1709 Rose Street
Berkeley, Ca 94703

HELP NEEDED INTERFACING N*DOS 5.2D TO SOL
by Robert R. Walling

Dear Mr. Sokolow:

I recently purchased North Star's new DOS5.2D/BASIC and thought I would be able to use the NorthStar Input/Otput routine for SOL Computer written by Joe Maguire, Feb 79, found in volume 2, number 3 issue of the newsletter. Ifound that I was not able to interface these routines successfully with the new DOS5.2D. I thought perhaps Mr. Maguire or some other SOL owner may have been successful in accomplishing this and would be willing to share the information. I would certainly appreciate any assistance that PROTEUS may offer. Thank you.

Robert R. Walling
9308 Edmonston Rd.
Apt-304
Greenbelt, Md.
20770

Sincerely,
Robert R Walling

Robert R. Walling

RING THE BELL ON A SERIAL DEVICE BEFORE PRINTING
by David L. Dalva II

I have recently acquired a Micro Complex 8Ø by 24 line board for my Sol. The installation is not hard. You start off by removing the mother board from the Sol's chassis in order to facilitate working and installing the new module. After this is done, some of the chips on the Sol mother board are removed. Next unsolder two connections to the Personally module. Solder in a small jumper connection. Plug in the 2Ø by 8Ø module. Plug in the new Personally module. Test the Sol before assembling it.

The board works well. In CP/M you can run a 58k system on an 24 by 8Ø screen which allows almost all of the CP/M software to run without extensive changes.

Micro Complex has also made use of the LOAD key on the Sol to boot the system at either CØØØH or FØØØH depending on the setting of the switch on the Personally module which also enables the screen for the correct size, 16 by 64 at CØØØH to run standard Sol or Ptdos or 24 by 8Ø at FØØØH for CP/M. They have also added some new commands to the Personally module including a memory test, a ASCCI dump, a fill command, enter ASSCI character string, a move memory block move, a compare memory block and other commands. They have eliminated the tape commands. They also sent a new listing of SOLUS which they have reassembled.

I have not had any trouble in installation or in operation. A phone call to Bob Hogg results in a prompt and knowledgeable answer.

I am enclosing a small program to ring the bell on a serial device before printing. It checks that the printer is on and that the local key is set correctly. I believe with the correct port it will work on a parallel port also.

I have been using PT's Business Basic in the DØØØH version and have found that it works very well in a business environment. The Common command allows the passing of values from one program to another. You are able to lock out the MODE SELECT key and to prevent the listing or stopping of a program in Basic. It also encodes the listing on the disk so that it will not make sense if printed. A Business Basic program will not run under regular PT Basic. Programs may be transferred by being saved in the Text mode which is readable by both versions of Basic.

Sincerly;

David L. Dalva II

```
 1Ø REM.   ************************************************
 2Ø REM.   *  Test printer before allowing printing       *
 3Ø REM.   *  Program name       :     PRINTS              *
 4Ø REM.   *  Created            :     28 Dec 1981         *
 5Ø REM.   *  Version            :     1.Ø1                *
 6Ø REM.   *  Last modified      :      4 Jan 1981         *
 7Ø REM.   *  Checked            :     DLDII               *
 8Ø REM.   *  Ref.               :     none                *
 9Ø REM.   ************************************************
 91 REM
 92 REM.               X4$   =  Name of report tobe printed
 93 REM
1ØØ SET CM=Ø: CURSOR 1,Ø: PRINT "": CURSOR 1,1Ø
1Ø5 PRINT "If Diablo sounded then printer is ready to proceed"
11Ø OUT 249,7: REM.                   Sounds bell on Serial device
12Ø CURSOR 2,Ø: PRINT "": CURSOR 2,2Ø: SET CP=1
125 PRINT " SPACE BAR ";: SET CP=Ø: INPUT (1,Ø)"  to continue    ",A$
13Ø CURSOR 1,Ø: PRINT "": CURSOR 2,Ø: PRINT ""
135 CURSOR 2,17: PRINT "NOW PRINTING : ";X4$
14Ø REM
15Ø REM.    If you have installed the Micro Complex 8Ø by 24
16Ø REM.    then the command "OUT 252,7" will cause the Sol's
17Ø REM.    bell to sound.
```

SD EXPANDORAM I MEMORY, 16KRA DELAY, ANOTHER SELECTRIC DRIVER
by D.A. Howe

I have promised for some time to send some of the information that I have acquired on the Sol Systems along to the readers of Proteus. Some of the comments in the latest issue (Vol. 4 #3/4) that I received today prodded me to action.

— SD Expandoram I Memory —

I have two of the Expandoram I memories that I have been using for over a year in two different Sol-Helios systems. SD Systems was little help in providing the correct jumper configuration, but once this configuration was established the boards provided good service. The only trouble I have had was related to dust in the IC sockets. Both boards were build from kits. I use the following modifications (refering to the pin numbers marked on the Expandoram I Board):

Connect Jumper	Cut Jumper
E32 to E33	E31 to E32
E19 to E20	E21 to E20
E2 to E3	
E9 to E18	
E5 to E6	
E7 to E8	
E10 to E11	E11 to E12
E15 to E16	
E22 to E24	E22 to E23
E25 to E26	E26 to collector of Q1
E28 to E30	
R10 (wire in place of R10)	
U25 Pin #3 to S-100 Pin #99	
U4 Pin #3 to U4 Pin #14	
	U4 Pin #3 to U4 Pin #5

It is of course always necessary to make sure all solder flux residue is removed from any home assembled board.

— 16KRA Delay Unit —

I have several of the 16KRA memory boards. Early on one of the delay units failed and I suggested to one of my students who was working with the Sol at that time that it could probably be replaced by a series of CMOS gates and he came up with a circuit using the 74C902 IC. The 74C902 CMOS non-inverting buffers were selected because each gate has an approximate progation delay of 50 nanoseconds when operated on a +5 volt supply. The delays of 100ns, 150ns, 250ns and 350ns can then easily be achieved by connecting two, three, five and seven of the 74C902 buffers in series. The delay was removed from the PC board and a Socket installed in place of it. Then the two IC were installed one on top of the other (piggy-back fashion) by bending and soldering the appropiate pins (and a few bits of wire for internal jumpers between individual buffers). This substitution has been in use over a year without any problems.

— Another Selectric Driver —

I have been using a Selectric typewriter that I purchased from Pacific Offic Systems of Palo Alto for some time. The Selectric unit they sell was originally installed in a GTE terminal. For this function GTE added a PC board to the bottom of the bare Selectric 725 mechanism and installed in their case. Pacific Office Systems bought the used unit, added their own PC board (still using the GTE addition) to provide an interface to a standard parallel port (I believe they also have a serial board) and an EPROM code converter from ASCII to Selectric code. This sounds complicated but it all works most of the time.

(The principal problem being that the Selectric mechanism requires frequent mechanical adjustments.) At the time I bought the printer I had a Sol3 driver on one of the system disks I had at the time. Sol3 is configured for a serial port. Only four statements need to be changed to produce a useful driver. These are:

```
        OUT F9  to  OUT FA (change from output to the  serial
                          to the parallel port)
```

This statement is located near the end of the Sol3 program. And five lines farther on:

```
        IN F8 to IN FA
        ANI 0A2H to ANI 4
        CPI 80H to CPI 0
```

After using this driver for a while I found the need for a more sophicated driver. (By this time I had modified the 725 mechanism with a 27 tooth index rachet gear to a 715, single pass film ribbon, with a 54 tooth index rachet gear and adjusted the index mechanism so that it would perform half line index under computer control; thus allowing exponents and subscripts to be printed from Word Wizard documents.) So I further modified the Sol3 driver to support underlining (by using the back space) and to pause (by typing a non-printing character, 5CH or 7CH) while the type ball is changed (to allow different fonts or math symbols to be inserted) and continue when the C/R is pressed. I will provide the program on disk for a future library disk if it is desired.

This letter has been on my letter disk for three months now and I have not found time to finish it. So I will send out what I have written and say that I am operating a real time clock system and an interrupt driven multitask system that I have implemented on the SOL-Helios system. I will be happy to answer inquires on the development of either of these items. For the next several months I will be in Jeddah, Saudi Arabia and can be reached at the following address:

Dr. D.A. Howe
Department of Applied Sciences
College of Engineering
King Abdulaziz University
P.O., Box 9027
Jeddah, Saudi Arabia

I would also like to hear from anyone else in Jeddah operating a Sol.

December 21, 1981

Sincerely,

D.A. Howe

SOL-20 BACKPLANE PROBLEMS?
by Bob Marsh

Are you having problems with intermittent system operation that goes away if you bnag the S-100 cards or unplug/re-plug them? This seems to be a common complaint of SOL-20 users, caused by the low spring tension of the TI connector used by PTC. (IMSAI and others had the same problem with these connectors.) Let's get together and have some new backplane PCB's made that will use AMP or Sullins connectors. I have access to the PCB artwork and it's ready to go. Estimated cost in small quantities $40. AMP connectors run about $3 to $4. Let me know via Proteus ASAP if you are interested in this, and we'll do it.

Bob Marsh
Proteus
1690 Woodside, St#219, Redwood City, CA 94061

H/Z-25 PRINTER

Equipment Review by Jeff Tom

After a long wait, I think I've found a printer I can stick with, the H/Z-25 Printer. A friend at work was the one who showed me the ad for the printer, initially. Almost immediately, I was certain that this was the printer for me. The price is $1095 for the kit, and $1595 for the assembled version. I opted for the kit, and with the $200 trade-in on my H-14 Printer, it made the price fairly reasonable.

The kit comes in two boxes, and if you get it through the mail, your mailperson will hate you forever, as it weighs 82 lbs. Heathkit followed its usual form of pre-assembling easily damaged components, so the only circuit board that needs to be assembled is the controller, the other 3 circuit boards are assembled (and tested?), thereby increasing your chances of having the printer work the first time. It took me about 16 hours to build the kit, and all but an hour or two was mechanical assembly. The kit is the same as the commercial version, so mechanical alignment and fit are excellent.

The printer consists of 3 major assemblies, not counting the cabinet. These are the Printer assembly, Paper Feed assembly, and the Electronics assembly.

The print head is what I have heard called a "ballistic head", so the heat problem associated with the H-14 is gone. The print head is positioned by a stepper motor, via a drive belt, which means head position control is positive and accurate. Signals are transmitted to the head via flat ribbon cables. An interesting touch. The printer uses a cartridge ribbon. It is easy to change, but I am not sure if there is a second source for this part.

The Paper Feed assembly has two tractor-feed sets. One above the print platen, and one below it. Both of these units are driven by a stepper motor and belt, so paper movement is good, both up and down.

The Electronics assembly consists of the power supply, and a board cage. These are air-cooled by a fan. There are 3 boards used. A Head Driver, a Motor Driver, and a Logic Board. The printer is controlled by a Z-80 u-processor, and has several K of ROM, and 1K of RAM, although only 256 bytes are used as the input buffer. LED's are used extensively on all three boards, so monitoring signals and status is quick and easy, if necessary.

Assembly of the printer was straight forward, and although a long process, it was easily accomplished. At light off, one wire of the print head would not operate. A quick look at the LED's showed me that the signal was being processed, and a continuity check showed me that the ribbon and print head were good. When I was re-assembling everything, I noticed that one of the ribbon cables did not appear to be seated in its connector cover very well. I pushed the cable into its cover, and everything worked.

The incident related above is a strong point in favor of owning this printer. If the unit goes down, you have all the schematics, diagrams, and descriptions, to aid you in repairing it, as well as store and factory parts support with no minimum charge. One of the drawbacks of owning this unit is its size. At 26" x 18", it is BIG.

Some of the features that I was looking for and found in this printer were; descenders, bi-directional printing, underlining capability, speed, and adjustable print pitch. The H/Z-25 has all that and more. Add double width characters, high speed line feed after 4 consecutive line feeds, limited graphics, handles forms from 3 1/2 inches to 17 inches, form feed control, variable line spacing, programmable horizontal and vertical tab settings to name a few more features. Oh yes, there is a "beeper" to get your attention too.

Interfacing the unit is by serial transmission using either the 20 ma. loop or RS-232 Interface. It is impossible to modify this unit like the H-14 as the DB25-P connector is soldered to the logic board. I had to make up a "null-modem cable" (REMEMBER: The SOL is an intelligent TERMINAL, so it looks for data on the same pins as the printer. The CTS and RTS lines, and Data IN and Data OUT lines must be criss-crossed.). I am using the same printer driver that I used with the H-14. Many of the printers features can be controlled by either switch setting or software. In the latter case, the printer recognizes ESCAPE sequences, or control codes to set up various things. Most of them do not effect the SOL, but some do. This makes a mess out of your screen at times, if your printer driver echoes output to the screen, as mine does, or if you imbed these codes in a text file using EDIT. The only thing left for me to do is to rewrite some of the custom routines for my text processor (POW, DDJ #29), to take greater advantage of the things the printer can do.

It remains to be seen if internally generated heat will affect performance and reliabilty, but for now, I am totally satisfied with my H-25.

There is one unused socket on the logic board that is reserved for future use, and I heard a rumor that there would be an add-on feature out this summer to give the printer "letter quality" print. Maybe...

In my opinion, I think its a darn good printer for the money, and worthy of your consideration if you are looking for a new printer.

Shown below are the printer's graphics characters.

MICROBYTE IN SAN JOSE HAS SOL PARTS FOR SALE

Microbyte Computer Systems
2798 South Bascom Avenue
San Jose, CA 95124 (408) 377-4685

We have the following for sale at very reasonable prices.

1. Sol-20 running no RAM boards - 750.00
2. Sol-20 , power supply bad diodes 700.00
3. 16KRA RAM board parts missing - 50.00
4. Micropolis single drive with controller model- plus cable, diskette and P.S. - 800.00
5. Format controller bare board-RevD FC301004 Assy No.301003 (one only) -15.00
6. two CUTS bare boards - 10.00 each.
7. GPM with Cuter ROM A&T - 50.00
8. Ius 8K RAM (2102's) without RAMS-works - 30.00
9. Tektronix T912 dual trace storage scope 10 MHZ BW with probe - 950.00

Call us or write us to order. Please enclose check or M.O.

CATALOGUE OF PROTEUS PRODUCTS AND SERVICES

April 1, 1982

(Note: All prices are in U.S. dollars.)
See ordering information.

SOL & S-100 HARDWARE

Item M1: McVideo 24x80 video upgrade for Sol. $295.00

Plugs onto main Sol board, 24 lines by 80 columns. Uses no S-100 slots. Audio beep generator. Fast RAM for compatibility with future processor speed-up conversion. Switches back to 16x64 for compatibility with old software. Eliminates screen snow, optionally vanishes out of RAM space. Requires new Solos or Dual Personality Module (see Item M2). NOTE: your present video monitor may not have adequate resolution (bandwidth) for the smaller characters, so you may need higher quality video monitor. Installation is not difficult, but takes some care and a little soldering.

Item M2: McDPM -- Dual Personality Module. $95.00

Replaces Solos personality module and has two EPROMs (2716). Switch on back of module (accessable without opening Sol covers) selects which EPROM is active. When used with the McVideo upgrade (item M1 above), the switch also selects Sol RAM, ROM, and video to be at C000 or F000 address, and sets video display form (16x64 versus 24x80). Programmed with various versions of Solos (see below).

The new McSolos is an altered version of Solos which replaces the tape I/O commands with other commands, such as Test Memory, Dump Ascii, Move Memory Block, etc. It also provides for a selectable auto-bootload on power-on/reset, or bootload on LOAD key. See description of McSolos in Proteus News, Volume 5 Number 1. (Custom versions available, request price quotation.)

Standard configurations:

Configuration A = 2716 EPROMs in both C000 and F000 socket, programmed with same version of McSolos except the C000 origin McSolos provides 16x64 video routines, while F000 McSolos provides 24x80 routines.

Select this configuration if you want to have identical Solos monitor features in the C and F modes. To use tape routines, you should load them from disk or replace your old personality module temporarily. (Note: a Processor Tech personality module will still work after installation of the Dual Personality module, but a minor change must be made to it first; explained in installation guide.)

Configuration B = Empty socket C000 and board jumpered to accept 9216 masked ROM in C000 socket; you move your original Solos ROM into this. Socket F000 has a programmed 2716 EPROM containing McSolos with video output routines for 24x80 screen.

Select this configuration if you want your Sol to function completely normally with the Dual Personality Module switched to C000 setting, including tape routines, but to talk to the 24x80 screen properly when Sol is relocated to F000.

Configuration C = unprogrammed 2716 EPROMs in both sockets.

Select this configuration if you don't plan to use the 24x80 video upgrade, can program your own 2716's, and want to relocate Sol's address space to F000 for more contiguous RAM space or if you want to do something special in Solos.

When ordering, specify which configuration and which disk controller bootload routine to include in the McSolos:
1. Helios controller.
2. NorthStar. Specify ROM origin and DOS origin.
3. Tarbell single density controller.
4. Versatile Disk Controller (Proteus item M5).
(Controllers that bootload by executing an address that ends in double zero, such as F800, will bootload using the auto-jump feature of the McSolos. Other controller bootload routines are available at extra cost.)

Owners of earlier version of the Dual Personality Module can have the EPROMs reprogrammed for use with the video upgrade for $15 each. To avoid loss of use of your machine during EPROM re-programming, you can send the programming fee plus a $25 deposit with the order, we will ship programmed EPROMs, you install them, and you send the old EPROMs back for refund.

Item M3: McFloppy Disk Controller. $450.00

A NorthStar work-alike, fast hard-sectored format, but better than NorthStar's. S-100 and Sol compatible. Up to 96 tpi DS, DD. All digital for reliability. Up to 816 K per mini-diskette with appropriate disk drive. One S-100 board. Onboard ROM relocates to any address, with selectable switch. The McDOS (tm) disk operating system included works like NorthStar's DOS. CP/M available from Lifeboat Associates for 48 track drives. Soon, a CP/M-work-alike for 96 track drives will be available.

NEW!
ITEM M4: SOL Keyboard Customizer & Maintenance Kit $35.00

A replacement for the Z18 keyboard encoder ROM on Sol's keyboard pc-board, plus 16 re-legendable keytops and spare parts for keyboard. Full instructions included.

The new ROM encoder sets the high order bit (bit7) when any of the 15 keys on the numeric keypad section are depressed, to distinguish them from the corresponding key on the main alphanumeric section. The other 7 bits are unchanged from their usual coding. All other keys function normally.

In some Sol's, installation of the ROM is merely a matter of unplugging the old one and plugging in the new one. In newer Sol's, some soldering is required.

The re-legendable keytops are for replacing the 15 numeric keypad keytops, plus one spare. They allow you to place your own labels on the keys, under a protective clear cover, thus defining them to be whatever functions you desire. Programs can interpret the keytops to have special meaning. In CP/M, your BIOS can identify the function keys by the high order bit being set. It should then substitute other characters for these special keys.

One suggested application: Relabel the keys to have commonly used word-processing function names. This way, you don't have to memorize the control-character sequences to do insertion, deletion, previous-page, etc. If you use several different editors, for example, let the BIOS translate your single key command into the appropriate code for the particular editor in operation.

CP/M users should investigate the SMARTKEY CP/M utility for defining special function keys described in Byte magazine, March 1982 page 477, available from FBN Software, 1111 Sawmill Gulch Road, Pebble Beach, CA 93953, (408) 373-5303. This routine intercepts the console input data and substitutes a user-defined string of keypresses for selected keys. For example, if you frequently access a dial-up network such as The Source, you can use one key to enter the whole sign-on access sequence. Installation of SMARTKEY requires little or no software expertise.

NEW!
Item M5: Versatile Disk Controller $495.00

In one S-100 board, this controller provides three major features: dual-density 8" floppy disk controller, Priam SMART bus hard-disk / tape interface, and real-time calendar/clock. Both disk controller interfaces use DMA to communicate with S-100 bus system. The board uses no address space.

The floppy disk controller comes configured to be a directly plug-compatible replacement for the Helios controller. That is, it interfaces to the PerSci 270 diskette drive as used in the Helios disk system. This allows Helios owners to use standard CP/M soft-sectored 8" diskettes (single or double density). A configuration header allows the controller to be adapted to many floppy drives, such as the Shugart. It will handle single-density and double-density IBM standard format, single or double sided 8" drives. Uses DMA and does not take up any system address space. (Note: Some Processor Technology old 16KRA and 32KRA memory boards, the ones with the long horizontal bar heat sink, will not work with DMA.)

The hard disk interface is plug-compatible with the Priam SMART bus, allowing use of any Priam hard disk and/or any other high-speed devices (such as tape drives) that use the SMART interface. Disk drives are available in 10, 35, 70, and 156 megabyte size. Tape drives are available from several vendors with SMART bus interfaces. This allows industry-standard tape drives (1/2" and 1/4") to be attached to the same cable as the disk for back-up capability. Controls up to 4 devices. Uses DMA and does not take up any system address space.

The real-time calendar/clock maintains time-of-day, day-of-week, and date. The clock keeps working even when the system power is off, through use of an on-board battery. The computer can read the time and date through I/O ports.

CP/M is available for this controller for $170, ready to run on a Sol-20. The CP/M disk includes a number of extra utilities, diagnostics, etc. Text editor with special command for reading time and date into the edited text is available. See software section of catalog.

Helios owners: it is easy to install. Unplug the Helios boards from the Sol. Unplug the disk drive cable from the Helios controller board. Plug cable onto the Versatile Disk Controller, plug controller into one Sol slot, insert a modified personality Solos EPROM in the Sol, and you're ready to bootload. Request price quotation for kit to upgrade a Helios to contain an 8" hard disk drive (10 to 70 megabytes) inside the same cabinet.

Request price quotation for Priam disk drives.

Versatile Disk Controller (VDC)	$495.00
Also needs the McDPM or Bootload ROM to provide bootloading.	
McDPM (Proteus item M2) with McSolos & VDC bootload	95.00
Bootload ROM only (specify 2708's or 2716) with Solos	25.00
CP/M 2.2 ready-to-boot in Sol-20 with VDC	170.00

Item M6: Helios safety retrofit kit $35.00

Capacitor and fuses, with complete instructions for fixing a design fault in Helios power supply that will in time burn out and could cause serious damage to the disk drive. See Proteus News vol. 4 number 3/4 for full explanation.

DOCUMENTATION

Encyclopedia Processor Technica (EPT)

This is a multiple volume collection of all of the documentation we have on Processor Technology products. The volumes and prices are described below. All of the manuals and updates we listed in the past are now discontinued items. It was just too hard to keep reproducing bits and pieces for people. Instead, all of the material and more is available in the Encyclopedia. As future ideas and improvements come up, we will be issuing more pages for the volumes.

See complete listing of chapters in Appendix I at the end of this catalogue.

Volume 1:	History and Basic Operation of the SOL.........	$35.00
Volume 2:	SOL Technical Manual............................	45.00
Volume 3:	Programming the SOL Computer....................	35.00
Volume 4:	Programming Manuals.............................	50.00
Volume 5:	ALS-8 Manual and User Notes.....................	30.00
Volume 6:	PTC Memory Boards...............................	45.00
Volume 7:	PTC Input/Output Boards.........................	40.00
Volume 8:	The Helios Disk System Manuals..................	35.00
Volume 9:	Helios Service Manual...........................	55.00
Volume 10:	Programming with the Helios Disk System........	60.00
Volume 11:	SolPrinter & Hytype Technical Manual...........	60.00
Volume 12:	Sol a la Carte...(to be published in near future).	

SOFTWARE

Item P10: Discontinued.

Item P11A: GAMEPAC 1 (Target, Life, Pattern, Zing games) including manual. On Sol/CUTS cassette, runs with SOLOS/CUTER in Sol-20 or S-100 system with VDM. $5.00

Item P12: Source code for Optional Precision Disk BASIC 1.1 mod 0, Disk BASIC/5 and BUSINESS BASIC.

License for personal use	$100.00
On 3 Helios disks (PTDOS format)--license required.	90.00

Item P13 Discontinued.

Item P14: Source code for ALS-8 program development system consisting of memory-to-memory assembler for 8080 microprocessor assembly language, video editor, and 8080 simulator.

License for personal use.	$50.00
On Helios disk in PTDOS format.	30.00

ALS-8 manual and users notes in EPT Vol. 5.

Item P15: Source code for PTDOS, the Processor Tech disk operating system, including most command files.

License for personal use.	$100.00
On Helios disks (3)	90.00

Item P16: Helios diagnostic programs source disk. $30.00
On Helios disk 45.00
Documentation appears in EPT Vol. 9.

Item P17: EDIT, cassette-to-cassette editor that allows editing of the source files on tape. It edits files in SOLOS/CUTER byte-mode (256 byte blocks), PTDOS CTAPE format, or Extended Cassette BASIC text (T) format. Will edit files that are too large to fit into memory, since it edits block by block. Also includes Pack and Unpack routines that change block-mode files to byte-mode files. (Useful for changing Solos/Cuter SAved files into 256-byte block files, and vice versa. For example, ALS-8 and BASIC source files.)

On Sol/CUTS cassette in eXEcutable form $30.00
Manuals in EPT Vol. 4

Item P18: ASSM cassette-to-cassette assembler that will assemble source files too large to fit into memory in one load. Works well in conjunction with EDIT above.
On Sol/CUTS cassette in eXEcutable form $ 30.00
Manual in EPT Vol. 4.

Item P19: Nevada COBOL compiler, a subset of ANSI-74 COBOL for CP/M and MP/M disk operating system. Requires 32K RAM and one disk drive. By Ellis Computing. Available on most disk formats. $149.95

NEW!
Item P20: Nevada PILOT.
An extended version of PILOT language, by the original PILOT designer: John Starkweather, PhD. Designed for computer aided instruction, including control of video cassette recorder and voice response units. Easy to program for data entry, medical history taking, programmed instruction and testing. Available in most disk formats. Requires 32K CP/M disk system. Specify which type of disk you have. $149.95

Item P21: PTC Cassette PILOT on eXEcutable cassette. $30.00
Manual is in EPT Vol. 4.

Item P22:
PTC Extended Cassette BASIC executable cassette. $30.00
Manual in EPT Vol. 4.

Item P23: DISKT
Helios disk test program for diagnosing Helios hardware problems. On eXEcutable cassette. $30.00
Explained in EPT vol. 8 & 9.

Item P24: SOLT
Sol test program for diagnosing hardware problems
On eXEcutable cassette $30.00
Manual in EPT Vol. 2.

NEW!
Item P25: Nevada EDIT.
The CP/M version of Processor Technology's fast video editor, in the style of the ALS-8, PILOT, and PTDOS editors. Will work with memory mapped displays such as the Sol or VDM, as well as most terminals on the market. Designed for programming, not word-processing, although it can be used with text output formatting programs. Features single key commands for scrolling, block moves, search and replace, settable tab stops, file insertions, etc. Customizes itself for terminal's characteristics by menu selection. Available for most CP/M disk systems.
$119.95

NEW!
Item P26: VDC Edit.
A CP/M-compatible editor resembling the PTDOS video-oriented editor, but having extra features, including command to read time and date into the edited file from the VDC calendar/clock. This is designed for use by programmers to keep track of the versions of programs. You can do this by insertion of date and time of last modification into remarks in the source file. See the VDC clock/calendar feature described in Proteus Item M5. This is a programming editor, not a word processor.

On single-density CP/M 8" soft-sectored diskette $119.95

SUBSCRIPTIONS AND BACK ISSUES

Item PN2:	Proteus News back issue set, 1979	$15.00
Item PN3:	Proteus News back issue set, 1980	$18.00
Item PN4:	Proteus News back issue set, 1981	$24.00
Item PN5:	Proteus News, current subscription, 1982	$30.00

(Foreign addresses, via airmail printed-matter, current subscription--US funds only--US $38.00)
(NOTE: Proteus News is a periodical and thus exempt from California sales tax. Prices include postage.)

SERVICES

Item S1: Disk updates $25.00
We will update your PTDOS to level 1.5 mod 1, or any Basic Software Group original disk (WordWizard, AccPac, MailMaster, MailSort) to the latest known revision level. See page 18 of Proteus News volume 4, number 1, Jan/Feb 1981 for latest revision numbers.

Item S2: Membership List (Partial) $10.00
Roster of Proteus members in your area (excluding those who have requested their name not be made public). Approximately 100 names and addresses in the zip code area of your choice for $10.00. We do not sell membership lists to non-members, nor do we sell the whole list.

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 1

History and Basic Operation of the Sol

CONTENTS

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 2

SOL Technical Manual

Contents

Sol Change Notices (PTC-Originated)

User-Originated Changes

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 3

Programming the SOL Computer

Contents

Page

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 4

Programming Manuals

Page

Contents

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 5

ALS-8

Page

Contents

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 6

PTC Memory Boards

Page

Contents

Volume 7

PTC Input/Output Boards

Page

Contents

Volume 8

The Helios Disk System Manuals

Volume 9

Helios Service Manual

Page

Contents

Volume 10

Programming with the Helios Disk System

Page

Contents

ENCYCLOPEDIA PROCESSOR TECHNICA

Volume 11

SolPrinter and Hytype Interface
Technical Manual

Contents — Page

ORDERING INFORMATION

1. All prices shown in catalog are in U.S. funds. Please do not send foreign funds.

2. California residents add 6% sales tax on all items except Proteus News and services.

3. Add shipping charges as follows:
   a. On Encyclopedia Processor Technica, add $2.50 per volume for 4th-class. Add $5.00 per volume for first class domestic delivery (U.S., Canada, or Mexico). For foreign postage, add sufficient amount to cover 5 pounds per volume (inquire locally for international rates from USA to your area) and we will refund any overpayment.
   b. Hardware (items with "M" item numbers) add 3% of total price to cover shipping, handling, and insurance to domestic addresses (US, Canada, Mexico). Add 6% of total price for air shipment to other foreign addresses.
   c. Subscriptions, back issues, and software ("P" item numbers), price includes shipping.

VISA and MASTER CHARGE are welcome. Send account numbers and expiration date.

CONTENTS OF BACK ISSUES

1979 VOL3 #1 — $15.00

CONTENTS

1979 VOL3 #2

CONTENTS

1979 VOL3 #3

CONTENTS

1979 VOL3 #4

CONTENTS

1979 VOL2 #5

TABLE OF CONTENTS

TABLE OF CONTENTS

****

1979 VOL2 #6

****

TABLE OF CONTENTS

1980 VOL3 #1 $18.00

1980 VOL3 #2

TABLE OF CONTENTS

In this issue:

1980 VOL3 #3

CONTENTS

1980 VOL3 #4

1980 VOL3 #5/6

TABLE OF CONTENTS

1981 VOL4 #1 $24.00

TABLE OF CONTENTS

1981 VOL4 #2

TABLE OF CONTENTS

1981 VOL4 #3/4

TABLE OF CONTENTS

1981 VOL4 #5/6

TABLE OF CONTENTS

HELIOS LIBRARY DISKS

Library diskettes are donated programs. They are generally in public domain, unless copyrighted by the author within the program itself. All disks are $35.00 each, but if you donate an acceptable program for the library (on diskette), you will get a $20.00 credit toward a library diskette.

H - 1

Documentation of Files on this diskette

SOLOS:S A copy file for standard and my extended version of SOLOS. The original source was obtained from the CP/M users group. Proc. Tech. may have rights to this program. My mods are public. Functions with or without extension PROM board in memory. Can be used for Standard SOLOS by conditional assembly. Uses files named SOLOSn:S. (Donated by Ron Parsons.)

SOLOS:D Further documentation of the files SOLOS:S, etc above.

COPYF:S Copies the files listed after command (separated by commas) from disk 0 to disk 1 preserving attributes. Requests permission to rewrite an existing file. (Donated by Ron Parsons.)

RELOC:S Relocation program from July 1977 Byte.

NFILES?S Prints the number of files on the unit "/u" (Donated by Ron Parsons.)

FSDISP:S Displays the free space map on the unit "/u" (Donated by Ron Parsons.)

REMNUM:S Removes line numbers (first five cols.) from named files. (Ron Parsons)

INTSEL:S Interrupt driven background Selectric driver. To be (?) described in PTCs ACCESS. (Ron Parsons.)

SFILES:S Displays a compact list of all files on the unit given as parameter "/u". If no unit given, uses default. Name stands for "Short Files". (Ron Parsons.)

SFILES Command image for the Short Files command above. Recognizes the "/u" parameter.

PRROM:S Standalone Cromemco Bytesaver prom-programmer (SOLOS) (Ron Parsons.)

REORG:S A PTDOS disk reorganize. Copies all files from unit zero to unit one. Does not rewrite existing files. (Ron Parsons)

COMPAR:S Compares the two PTDOS files named listing differences Assembly source file. (Ron Parsons)

SOL:S Assembly source code for PTDOS command "SOLOS" which turns control over to SOLOS. Assumes SOLOS is at C000 as in the SOL. Qnce in SOLOS, the command "PT" will get back to PTDOS, assuming it is still unharmed in memory. (Stan Sokolow)

DSTAT EDBASIC program for descriptive statistics (mean, etc) (Stan Sokolow)

PRINTER Driver for selectric terminal (IBM 2741 compatible) on SOL's serial port. Uses SOL built-in RAM and PTDOS driver area. BE SURE TO CHANGE TYPE TO "D" BEFORE USING! Output device only. (Input not implemented.) System reset will wipe out initialization part of driver, so be sure to load new image of it. (Donated by Stan Sokolow.)

DMOVE:S Assembly source code for a delimited-move subroutine. It moves bytes from a source address to a destination address until count is reached or a delimiter byte is encountered. Unlike PTDOS's PSCAN routine, DMOVE lets the user define his own set of delimiter bytes. Additional explanation is in the code's remarks. (Donated by Stan Sokolow.)

LD List directory in alphabetical order, file names only. DOESN'T READ PARAMETERS; ALWAYS USES DEFAULT UNIT. (Donated by Chuck Ellis.)

S Jumps to SOLOS but gives description of all SOLOS commands first. Adds custom commands to get back to PTDOS. (Donated by Chuck Ellis.)

NEWGET This is a corrected version of the GET command that was originally released in PTDOS 1.4.0. It automatically will GET device files without RETYPE'ing them by hand. You can copy NEWGET to GET on your working diskettes. (Donated by Processor Technology Corporation.)

NEWFILES A corrected version of the FILES command which recognizes Upper and Lower case letters as equivalent in file names. The FILES released in PTDOS 1.4.0 treated the two cases differently when searching for files that match bracketed substring specifications. Eg., FILES s> and FILES S> found different files in the original version, but this version is corrected. (Donated by Processor Technology Corp.)

FOUR Generates random "four-letter words". Mode Select terminates and returns to PTDOS. Words are displayed in large block letters on the video screen. Rated PG --Parental Guidance recommended.

HELP Provides a brief explanation of PTDOS commands. If a command file name is given as the argument after the name HELP, an explanation of the named command will appear. Otherwise, a summary of the HELP command is given. (Donated by Processor Technology Corp.)

HELP:D This is the reference data for the HELP command. HELP expects this file to be on the default unit.

MIND:S Assembly language source for the MIND Robot Control Language by Lichen Wang, see DR. OOBBS JOURNAL, Sept 77, revised by Ken Anderson, DR. DOBBS, May 78. (Donated by Earl Herr.) IN ALS-8 FORMAT <A>.

PASCAL The Stanford Micro Pascal System, dated 9-13-78, from Stanford Linear Accelerator Center, Stanford University. (Donated by Sassan Hazeghi, Computer Group, S.L.A.C.) Essentially the entire P-code implementation of the PASCAL computer language, as implemented for the IBM 360/370 computers. Except for generalized FILE declarations and passing FUNCTIONS/PROCEDURES as parameters, it adheres to the standard PASCAL as defined by Jensen and Wirth in the 1974 PASCAL User Manual and Report. It is NOT the U.C.S.D. system. The Stanford version runs under PTDOS, and thus it can pass data to other programs not written in PASCAL using normal PTDOS file structures. This preliminary release does not have the REAL arithmetic implemented in the interpreter, so only 16-bit integer arithmetic can be used, even though REAL will compile. We expect to have the REAL arithmetic in a future library release.

See file PAS.DOC for a more complete description. The source PASCAL for the compiler, post-processor, and the assembly source for the interpreter are NOT on this diskette. SOURCE CODE IS ON NEXT VOLUME OF THE LIBRARY.
TO RUN THIS SYSTEM YOU SHOULD HAVE 48K CONTIGUOUS RAM FROM THE BOTTOM UP, TO HOLD PTDOS AND THE PASCAL SYSTEM. MORE MEMORY CAN BE UTILIZED, BUT IT TAKES MODIFICATION TO THE INTERPRETER. A VERY LARGE PROGRAM (THE PASCAL COMPILER ITSELF) WAS COMPILED IN THAT WAY ON A SOL. See the interpreter source code for the instructions on modifying it for 64K memory.

PAS.DOC Documentation for the Stanford Micro Pascal System.

THE FOLLOWING FILES ARE RELATED TO PASCAL AND ARE DESCRIBED IN "PAS.DOC": PAS.S, PASM.S, PINTRP.S, PASCAL, PASM, PINT, COMPILE, RUN, TEMP.T, TEMP.P, QUEENS:S, SORT:S, XREF:S, SOMA:S POBJ, F:S, PAS.CMPL, PAS.DEFS, INITPATB.

FOOTBALL An EDBASIC program for 2 player video football. Self-documenting. (Donated by Gerry Fricke; adapted to disk BASIC by Stan Sokolow.)

NOTICES Important legal notices regarding this diskette.

WARRANTY The limited warranty on this diskette.

FEEDBACK Explanation of how to report problems you encounter with this diskette's programs.

CONTENTS OF PROTEUS DISK H2

PAS.S Source code for the SLAC Pascal compiler, written in Pascal. Read the file PAS.DOC on Proteus disk H1 for background information on the compiler. Also see Proteus News, Vol 2, No 2 and following issues for more details on SLAC Pascal. The object file from this program is on disk H1, along with all the supporting programs to compile Pascal programs of moderate size. Compiling PAS.S requires the 64K version of the SLAC system. Disk H1 has the 48K version. See INTRP.S below for information on creating a 64K version.

PASM.S Source code for the post-processor (assembler) of the SLAC Pascal compiler.

RUN.S Source code in 8080 assembly language for the RUN command that is described in PAS.DOC on disk H1.

COMPIL.S Source code in 8080 assembly language for the COMPILE command described in PAS.DOC on disk H1.

INTRP.S Source code in 8080 assembly language for the run-time interpreter that executes the compiled SLAC Pascal object programs. This interpreter simulates a pseudo-computer whose "machine language" is known as "P-code". Read the comments in the code to see how to reconfigure the interpreter to run in more memory space. The more space the interpreter has for the P-code, the larger the programs it can run. To compile a program as large as the Pascal source code PAS.S, you must reassemble the interpreter to utilize all available space in a 64K Sol. PLEASE NOTE: This is still only a preliminary version of the interpreter. Although the compiler will produce P-code for floating point arithmetic (REAL variables), the present interpreter will not execute these P-codes because the floating-point package has not been implemented. For the present, just avoid REAL variables. In future volumes of the Proteus library, we expect to have an updated version which implements floating point operations.

INTRPn.S where n=1,2,3,4. These are the portions of the interpreter for SLAC Pascal P-code. INTRP.S copies them into the input stream of the PTDOS assembler ASSM. They were divided this way just for ease of editing with EDIT.

SORT.B A sorting program for sorting a data file. This program is written in EDBASIC and uses the Shell sorting method. The READ statement will need to be modified to fit your file's data. The key items which the sort looks at to place the record in order can be modified. See the REM statements in the program.

SORT.F Similar to SORT.B, but written in PTC FORTRAN.

DSORT An EDBASIC program to create data files for SORT.B

DRAGON DRAGON ISLAND game. Hunt the dragon in his dark caves and win the hand of the princess.

WARRANTY The limited warranty on this diskette.

NOTICES Important notices regarding this diskette.

FEEDBACK How to report errors you find.

CONTENTS This file.

H - 3

1979CAL The 1979 calendar.

AUTO.SYS, AUTO, RATES, RATES-C, CHANGE, R.FILE-S, R.FILE-C

An insurance agency software package, for rating CSL or split limits automobile insurance. Read text file AUTO.SYS for complete explanation. [Martin Hill, Jr.]

SETDATE, DATE?,DATE:S

A `SAFE' program to set the date in PTDOS global area, designed to be placed in START.UP primarily. DATE? similarly fetches the date from memory for checking. SETDAT:S and DATE:S are their respective source codes.

HEAD

A header program. Before you list your program, you will want to XEQ HEAD. HEAD will print out the name of your program, ask you to GET and then LIST the program you want. At the end of your listing, XEQ E and you will now be back at video control. The SET OF= command is on line 88; change it to have the name of your output driver. Mine is PRNT1 for a Centronics 701 and PRNT3 for an old communi- cations printer. [Donated by Roy Heybrock.]

HGRAPH:S

HGRAPH:S is an extended basic bar-graph program which includes 4 functions used in plotting numeric bar graphs. An array is passed to the function to produce the bar graph. Included are functions to create both axes plus an overlay grid. Also histograms may be drawn by passing the median value to the plot function along with the array.

HYTYPE

PTDOS driver for Diablo HyType printer using Processor Tech's HyType interface board and plugged into the parallel port of the Sol. This driver is WordWizard compatible.

It assumes a Courier 10 printwheel, or equivalent arrangement. It is based on the SolPrinter2 driver, with references to the extra features of the SolPrinter interface deleted. [S Sokolow]

MEMTEST

A 48K memory test. Test lower 48K memory. Puts BEL character out after each cycle. To quit, reset system and re-boot.

MTEST:D

Documentation of MEMTEST.

OKIDATA

Source code for PTDOS driver to operate Okidata printer on serial port of Cromemco TU-ART interface board.

PRNTi

Centronics 701 driver. [Donated by Roy Heybrock.]

PRNT3

Driver for an old communications printer. [?]
[Donated by Roy Heybrock.]

S154C

PTDOS & WordWizard driver for a Selectric terminal IBM 274i-type, Correspondence coded, on the Sol serial port, modified to do 134.5 baud. [Stan Sokolow]

RETIR

An updated version of RETIR on H1 disk to change graph headings that were not set up right. [Donated by Roy Heybrock, CFP.]

MASTERMD

This is a dual mastermind game. Two people play at the same time on two different game boards. The game features a none repeat of numbers within the matrix, automatic return, and a more than one game set to determine the winner. Each play enters his or her try as part of one large entry. (i.e. The first player enters a guess and a comma. The second player writes his guess on the other side of the comma and the machine automatically continues without the need for the carriage return. [Michael Richardson]

CAL%%%

I teach Mechanical Technology at Saunders Technical High School in Yonkers. This program will print Mechanical Technology - as easy as Pi on the top of a calendar. What's nice about the program is that the calendar is generalized so that any year in the 20th century can be entered and an accurate calander will be produced. There is also an interesting generalized, large print, printing of the year. (This is what is taking the time between the picture printing and the calendar printing) [Michael Richardson]

S0, S1, S2, S3, S4, S5

My start up programs. [Michael Richardson]

CALENDAR

A program to give the day of the week and the number of days old you are. It also gives the Julian day number for those of us who are into Astronomy. [Michael Richardson]

MULTABLE

A multiplication practice program. [Michael Richardson]

SUMTABLE

An addition practice program. [Michael Richardson]

W1

A program that I use whenever I go on a diet. The data is recorded in data steps at the end of the program. Each time a new day is entered the program expects the data to be the day number, the food, the calories, and the weight at the beginning of the day. (The weight is omitted for all other entries on that day.) There are alphabetized calory counts for some of the foods already written into the program in REM statements. [Michael Richardson]

W2

A graph drawing program that I use whenever I go on a diet that is drastic enough for me to use program W1. [Michael Richardson]

TT

The program I use to automatically right hand justify a letter. [Michael Richardson]

T

The program I use to print a file I've already right justified. It is info protected (I attribute). [Mike Richardson]

CMPF.A

File compare command, in assembly source code. See the program comments. [Tom M. Quinn]

MEMBNK.A

Memory bank-select, Cromemco-style. See the source code. [Tom M. Quinn]

BAUD.A

Command to select baud rate on 3P+S board. [Tom M. Quinn]

LOOK.A

Command to find specified bytes in memory or in a disk file. [Tom M. Quinn]

SETPRT.A

Command to initialize TI 810 printer. [Tom Quinn]

TI810.A

TI 810 printer driver based upon Sol Printer 3 driver. [Tom M. Quinn]

SORT.F

Shell sorting in PTDOS-FORTRAN. [Earl Dunham]

SORT.B

Same as SORT.F, but written in BASIC. [Earl Dunham]

DSORT

A BASIC program which creates data files for SORT.B [Earl Dunham]

PROCESS

A video-type editor to be used with the EDITOR program. To learn how to use PROCESS, run the BASIC program EDITOR. When it asks for the file name, enter PROCES:D. That file contains commands to PROCESS and also doubles as an example as to the usage of EDITOR. [Al Smith]

EDITOR

EDITOR:D contains information on using the commands available within the basic program EDITOR. To use EDITOR:D you must run EDITOR from basic. When asked for file name input EDITOR:D. [Al Smith]

LETTER

A letter from Al Smith regarding PROCESS and EDITOR.

L

Device driver for parallel printer. [Michael Richardson]

PROGRAMS

Explains the programs submitted by Michael Richardson.

STARTERS

Explains the starter programs by Michael Richardson.

EDIT:D

A brief explanation by Michael Richardson of why he renamed the PTDOS editor from EDIT to E. It also has a brief synopsis of the EDIT features.

BOOT:S The source for the Helios bootstrap loader.

COPYCA:S A screen doodling program which places a duplicate copy in memory for saving. Good for designing playing fields for video games.

DSKPORTS Equates for the Sol ports serving Helios
Gives port definitions and bit equates for all the ports used by the Sol for comunicating with Helios.

DOSIO:S

An I/O routine for the Sol for use with the North Star Microdisk system. Includes many desirable features.

8KRA:S Memory test programs:
16KRA:S
16K-1:S
32KRA:S
32K-1:S
48K-1:S A collection of memory test programs tailored
MTEST:S for the various boards produced by PTC. [J. Maguire]
[Editor's note: This MTEST is not same as MTEST:D on this disk]

OCTAL:S An Octal Enter and Dump routine. Gives ASCII values and a Hex address for reference.

PABAS:S Is the source code for Lichen Wang's Palo Alto Tiny Basic from early issues of Doctor Dobb's Journal. A few enhancements in this version.

SPINWR:S NEC Spinwriter 5510/5520 device driver.
A bi-directional, logic seeking printer driver fully compatible with WordWizard. It includes "space averaging"- an ability to restructure the line producing "type set" quality printing.

TERM:S The missing TERM command from Sol Bootload prom.

ZAP:S Zap let's you defeat the attribute protection of PTDOS. UNZAP:S gets you back.
Use with caution!!

The New and Better H - 4

PATCH1.5 is a program to allow the reatributing of attribute protected files on PTDOS 1.5 (not 1.4). See text file HOWDY for more information on operation.

MESSAGE,
!MESSAGE,
BUILD,
MES.S,
MES.TEMP are a group of programs that were done for fun. Studying the source code is a good way (maybe) to understand how to use the overlay handler and how to interface to PTDOS in general. See text file HOWDY for more information on operation.

FORMAT is the text formatter originaly described in Software tools, by Kernighan and Plauger. This version was written by Mike Gabrielson and printed in the May 79 issue of Dr. Dobb's. See text file HOWDY for more information on operation.

PRIMES is a fast program for generating prime numbers. See text file HOWDY for more information.

FACTOR is a program to factor an integer into it's prime. See text fila HOWDY for more information.

KWIKSORT is a quiksort or partition-exchange sort. It is neat in that it utilizes user-definable multi-line functions recursively, with automatic stacking of local variables.

QUIKSORT same as above but with modifications suggested by Knuth.

HEAPSORT from Knuth

SHELSORT from Knuth

SORT is another sort suggested by Knuth which seems to approach the speed of the quiksort but may not have the diadvantages(when the file is in order, for example) Knuth rates the mathematical evaluation of this sort at 50 points, his maximum.

MAZE is another example using recursive functions. The object is to find the longest possible word in a given matrix of random letters. The words are allowed to twist and turn as much as necessary but must not use the same letter twice. Try words like: location, tatterdemalion, pharmacutical to see it work. (Should be entered in lower case)

FIND+ was originally inspired by the IEEE micro-mouse contest. This was the best program I came up with, and it naturally uses a recursive function. Note that it will require lots of memory to run! SET your BUffer = 9000h and use the MBASIC on this disk.

FIND another mouse program but with a unique idea that causes interesting behavior sometimes. Consider an array where the walls are values at 999 and the corriders are set at 0 and the "cookie" is set at -1. Have the mouse increment any location he's at by 1 and then go to the lowest adjacent number. It words but looks funny sometimes. Also takes much less memory than FIND+. Watch its behavior in the top right corner.

PERMTEST-generates all the permutations of a given array in order. Makes a good problem. This algorithm from Dijkstra.

ACCOUNTS is a sample list of accounts used with BANKERS.

BANKERS is a program to enter deposits and withdrawals, and obtain readout of total and percentages.

CREATFIL will create random file of accounts. Used with BANKERS program.

EXPENSES is a sample list of expenditures for use with BANKERS.

PERSONAL is a program to create serial file of names or titles which may then be converted to a random file if desired using SER.RNDM program. For use with BANKERS.

SER.RNDM is a program to convert serial files to random access files for use with BANKERS.

gameDOC is some documentation on the game programs that follow:

ELIZA...CHASE...AMAZE...MMIND...BIORYM...LUNAR.1

EDIT+H19 is a short program to allow the use of the Heath H19 terminal with the PTDOS editor to use the terminals special function keys.

NUMSTR,
ADD.WK See specific documentation on these two programs.

QUME is a driver and source code which allows the QUME SPRINT 5 printer to operate at 1200 baud. Has build in handshaking.

SPO,
SPB are simple serial-port output drivers. Their source codes are SPO.S and SPB.S, which may be reassembled, etc. through MAKDRIVR macro.

TENSORTS compares ten sorting algorithms in EDBASIC, including three versions of the Shell-Metzner, quicksort, heapsort, plain and Woodrum merges, delayed-replacement, selection and bubble.

w

CONTENTS OF THIS DISKETTE H5

This diskette contains the source and/or object code for a few device drivers. Those with type IW are WordWizard-compatible printer drivers. The source code for these begins with the lower case "w", as in "wSol3". To use the object code, GET the driver onto your system disk, RETYPE it type "D" for driver, and then use it.

The SolPrinter drivers (mSol3, wSol3, mSol2, etc.) were written by the Basic Computer Group, Ltd., in Vancouver, B.C., for use in WordWizard, MailMaster, etc. They may be used with any software, but be careful where they load to be sure they don't overlap other programs. The source code may be re-ORG'd to other locations. The programs beginning with "m" meet the specifications of MailMaster and the AccPac programs. The drivers beginning with "w" are for WordWizard and support bidirectional logic-seeking printing in a foreground/background mode. This is described in PTC updates specifying requirements for WordWizard drivers. They may be used by PTDOS in general, but will only give the special features for WordWizard if set in word-processing mode by a control/status call.

The Sol2 is a Diablo Hytype II printer metal printwheel printer, interfaced to revision E Sol parallel port using the PTC interface for the printer. If you have this printer but a revision D Sol, you will need to make an adapter cable that reverses one set of data lines as described in the Sol manual. This driver assumes you have the revised Hytype interface which was named the SolPrinter interface. The original Hytype interface did not support some of the status conditions, such as paper-out, so the status test in the driver will need to be modified.

The Sol2E is the same as Sol2, but using the plastic printwheel Diablo.

Hytype driver is a Sol2E driver modified to support the original Hytype interface designed by PTC, not the later SolPrinter one.

Sol3 is a driver for a serial printer attached to the Sol serial port. It was designed for the SolPrinter3 which was a Diablo 2300 matrix printer, but it has also been used successfully with other printers, including the Epson MX-80.

DEC is a DecWriter driver for the Digital Equipment Corporation Decwriter. It can easily be modified for most common printers on the serial or parallel port.

XEROX or 1610 or WPXER are drivers for the Diablo/Xerox 1610 or 1620 daisy-wheel terminals. They support ETX/ACK protocol and bidirectional printing.

CDC is driver for Control Data 9317 matrix printer.

SPIN is for NEC Spinwriter.

TI810 is for Texas Instruments 810 printer.

If you modify these programs and reassemble them, please remember that PTDOS requires all drivers to be loadable as one logical block. The assembler doesn't usually create image files in this format. To convert the object file into a single-block image file, use the command:

```
 EXTRACT file,S
```

which will "scrunch" the file into one block if possible. The scrunched file can be RETYPE'd into type "D" for driver. The physical blocksize doesn't matter, but you should chose a blocksize for efficiency. See the PTDOS manual for more info on this matter.

YOU CAN DISCOVER WHERE THE DEVICE DRIVERS LOAD BY RETYPE'ING THE FILE TO "I." AND GIVING THE COMMAND:

```
 EXTRACT filename
```

Don't forget to RETYPE it back to "D". Object files that aren't of device type "D" can be EXTRACT'ed without RETYPE'ing.

CONTENTS OF H-6 DISKETTE

Special Note: The contents of this diskette are essentially the work of one author: Stephen Maguire. Steve is a 19 year old engineering student at the University of Arizona at Tucson. At the urging of his dad, Joe Maguire, Steve is placing these programs at the disposal of Proteus members, for their own use, rather than offering them for sale. (However, all rights are retained. Some programs may be offered for sale in the future to the TRS-80 market.) Steve's only request is that if you make or have suggestions for improvements, please contact him at the address given in the listings.

STRIP:S — A Basic program which will remove all REMarks from another Basic program which is stored on a PTDOS disk.

CONVRT:S — This Basic program will convert ALS-8 text files to the PTDOS format. This includes the removing of all line numbers. The program can also convert PTDOS text files back to the ALS-8 format. All line numbers will be restored.

STORY:S — An Extended BASIC program to solve the problem of the three shipwrecked sailors and their monkey. (This problem was given as a term project in a Basic programming course.) Can be modified to solve for any number of sailors.

RAIDER:S — An assembly language file that copies RAIDRn:S into the assembler input stream.

RAIDRn:S — (in 3 parts) The SOURCE code for the Space Invaders video game. This is the video game which the Japanese spent over 600 billion yen ($2,730,000,000.00!!!) playing in thousands of bars and coffee shops all over Japan. It is identical to the original arcade version with the exception of sound effects. (The author was involved in programming the original) This program alone is worth more than the price of this disk!

MEDIT:S — An assembly language file that copies the 5 part file MEDITn:S into the assembler input stream.

MEDITn:S — An object code editor (a machine code or memory contents editor as opposed to a text editor) with all of the features of EDIT plus special ones such as Mode Toggle (Ascii/Hex) etc. Great for examining or patching programs for which you do not have the source code.

MEDIT:D — Documentation for MEDIT

PAS.IO:S — Sol I/O routine for NorthStar Pascal Version 1.0 It recognizes the GOTOXY feature so that no user GOTOXY routine need be written. It contains such features as a print toggle. Typing control/p sends the output stream to PRINTER: instead of to CONSOLE: Another control/p flips it back.

STARWR:S — The source code for a Starwars video game. Shoot the Imperial TIE fighters before they get you. May The Force be with you!

COLSNn:S — (in 2 parts) The source code for the Collision video game. With 16 playing fields, 3 difficulty levels and a demonstration mode. Great for kids!

SPINWR:S — An improved version of the Spinwriter printer driver SPINWR:S which was on User disk H3.

SPIN:D — Updated documentation for SPINWR:S.

SPINUP:D — A summary of the updated features of SPINWR:S and SPIN:D found on this disk.

SPIN — A device file assembled from SPINWR:S.

BIOPLT:S — The complete source for a BASIC biorhythm plotter. This program is too large to be run on a standard Sol with only 48K of memory. BIOPLOT below is a version with all REMarks removed which can be loaded and run under PTOOS Extended Disk BASIC.

BIOPLOT — A compacted version of the biorhythm plotter. This can use the WordWizard print drivers if the Basic is initialized with the matrix operations deleted. This program can plot to the screen or to any width printer, as well as to text files on the disk. Complete user control over EVERYTHING!

BIOTEXT — This is a text file used by BIOPLT:S/BIOPLOT. This file contains in depth descriptions of all the commands in the biorhythm plotter. Other pertinent text is also contained in this file.

Z80CVT:S — A Basic program that converts 8080 assembly language source files into TRS-80 type Z80 mneumonics. Now you can write programs for the lucrative TRS-80 market using your PTDOS editor!

SIMUL:S — Solves simultaneous equations.
SIMLOD:S — Saves equations in the data file below.
SIMUDATA — The data file for holding equations to be solved.

H - 7

This file lists the contents of this diskette, H-7 from the Proteus Library.

This diskette contains programs which were donated by a number of people. I feel that due credit should be given to the authors and have therefore grouped the programs by author.

For further information please read the files WARRANTY and FEEDBACK.

Charles L. Athey, III
Proteus Librarian

The following were donated by Frank J. Sanders. These programs handle personal finances.

ACCOUNTS - SAMPLE LIST OF ACCOUNTS

BANKERS - PROGRAM TO ENTER DEPOSITS AND WITHDRAWALS, AND OBTAIN READOUT OF TOTAL AND PERCENTAGES

CREATFIL - CREATE RANDOM FILE OF ACCOUNTS

EXPENSES - SAMPLE LIST OF EXPENDITURES

PERSONAL - PROGRAM TO ENTER DAILY OR WEEKLY EXPENDITURES,AND OBTAIN READOUT OF TOTALS AND PERCENTAGES

SERIAL - PROGRAM TO CREATE SERIAL FILE OF NAMES OR TITLES WHICH MAY THEN BE CONVERTED TO A RANDOM FILE IF DESIRED USING SER.RNDM PROGRAM

SER.RNDM - PROGRAM TO CONVERT SERIAL FILES TO RANDOM ACCESS FILES

Here is a group of programs from Preston Briggs of Interactive Computing.

PATCH1.5 is a program to allow the reatributing of attribute protected files on PTDOS1.5 (not 1.4). I don't have the source code unfortunately but it is pretty simple. Type PATCH1.5 and it will execute and return to PTDOS. Then REATR the file(s). For safety, you should re-boot afterwards as PTDOS will continue to ignore the attribute protects.

MESSAGE, !MESSAGE, BUILD, MES.S, and MES.TEMP are a group of useless programs I did for fun. Studying the source is a good way (maybe) to understand how to use the overlay handler and how to interface to PTDOS in general.

MESSAGE will type a random (almost) message on the screen whenever run. I use it in my START.UP file to avoid the same old boot-ups over and over.

!MESSAGE is a utility file that contains the 32 messages that MESSAGE may choose from. !MESSAGE should be on the default diskette.

MES.S is the source code for MESSAGE.

BUILD is a DO file that I use to create or replace messages in the utility file !MESSAGE. It expects to run on the default disk and requires !MESSAGE and MES.TEMP.

MES.TEMP is the source file for a message.

FORMAT is the text formatter originally described in Software Tools, by Kernighan and Plauger. This version was written by Mike Gabrielson and printed in the May 79 issue of Dr. Dobb's. I added the necessary interfacing to work with PTDOS. Mostly what I'm donating here, is the typing effort. I don't think it violates anything and Gabrielson includes no copyright message so I assume it is for general use. I did not include the comments when I entered the code so one should reference Dr. Dobbs and Software Tools for help with the program. Type FORMAT sourcefile, outfile. Enter #1 in outfile to run to screen. FORMAT.S is the source file. TEXT is a sample file to be formatted.

Extended Disk BASIC programs:
(actually, these should all run on extended cassette BASIC too)

PRIMES is a fast program for generating prime numbers. The algorithm is from a fairly recent CACM article by Gries. I'm sorry I don't have the date. The program could be extended by using PEEKs and POKEs instead of an array to represent the sieve as each element in the sieve can have only two values.

FACTOR is a program to factor an integer into it's prime components. It utilizes the same algorithm as the PRIMES program and could be extended in the same way.

KWIKSORT is a quiksort or partition-exchange sort. Is neat in that it utilizes user-definable multi-line functions recursively, with aoutomatic stacking of local variables.

QUIKSORT same as above but with modifications suggested by Knuth.

HEAPSORT from Knuth

SHELSORT from Knuth

SORT is another sort suggested by Knuth which seems to approach the speed of the quiksort but may not have the disadvantages(when the file is in order, for example) Knuth rates the mathematical evaluation of this sort at 50 points, his maximum.

MAZE is another example using recursive functions. It was inspired by a contest a friend entered in which the object was to find the longest possible word in a given matrix of random letters. The words are allowed to twist and turn as much as necessary but must not use the same letter twice. Try words like: location, tatterdemalion, pharmaceutical to see it work. (Should be entered in lower-case)

FIND+ was originally inspired by the IEEE mico-mouse contest. This was the best program I came up with, and it naturally uses a recursive function. Note that it will require lots of memory to run! SET your BUffer = 9000h.

FIND another mouse program but with a unique idea that causes interesting behaviour sometimes. Consider an array where the walls are valued at 999 and the corriders are set at 0 and the "cookie" is set at -1. Have the mouse increment any location he's at by 1 and then go to the lowest adjacent number. It works but looks funny sometimes. Also takes much less memory than FIND+. Watch its behaviour in the top right corner.

PERMTEST generates the all the permutations of a given array in order. Makes a good problem. This algorithm from Dijkstra.

The following program was donated by Larry McDavid of LMC Engineering, Anaheim, Ca.

LOADM - PTDOS image-file load to memory. This program reads PTDOS image-type files into system memory starting at a user-selected address. The image-file block headers are used to control the loading of each file block so that the final loaded format is identical to that resulting from entering the filename as a command. The source is LOADM.C, and the documentation file is LOADM.D.

The following program was donated by Ben C. Stapleton Jr. of Office Supply Inc., Portsmouth, Ohio

PHONUM:S EDBASIC program converts Phone Numbers to Words. Each Phone Number generates 2187 different words.

The following programs were donated by Jay Parsons of Somerset Data Systems, Inc. Bernardsville, NJ.

ROBOTS is an old game involving hiding from killer robots.

TENSORTS compares ten sorting algorithms in EDBASIC, including three versions of the Shell-Metzner, quicksort, heapsort, plain and Woodrum merges, delayed-replacement, selection and bubble.

The following program to help convert between CP/M and PTDOS format files was donated by Gib Zeratsky, GreenLake, WI.

CPM-TXT Documentation in file CPM-TX.D

The following programs were donated by Earl J. Dunham of La Habra, Ca.

Weekly Reporting programs: ADD.WK, ANYMO, NUMSTR, WK.DOC

CONVERT : a comprehensive english<=>metric convertsion program.

STR-SORT : an unusual way of sorting strings, using the Shel-Metzer sort algorithm.

HELIOS Library Disk H-8
April 16, 1981

This disk contains the small C compiler as implemented by Ron Cain and inhanced by Ed Hirgelt.

```
Cc     - The running compiler/
Cc?.c - C source for the compiler.
Cc?.a - 8080 ASSM source for the compiler, the results of
         compiling the compiler.
C80LIB.A - The runtime support package for the Sol-Helios
         system.
CcDef - The common definitions needed by the compiler.
Cc.Txt - A description of each routine in the compiler.
Cc.Use - A short description on how to use the compiler.
Setup.a - An assembly routine which the compiler generates
           a call to to setup the C environment.
Test.c  - A sample C program.
```

Please address any questions to Chuck Athey (415) 449-8337, 5571 Shorehaven Circle, Livermore, Ca 94550.

PROTEUS CATALOGUE TABLE OF CONTENTS

A SIDE BENEFIT TO ELECTRIC PENCIL VERSION SS
by H. Leon Winter

I have some more mods to Electric Pencil version SS. This new one allows control characters to be placed in the text buffer. I've been wanting to do this so I can cause certain things to happen in a printer handler. A side benifit came by surprise. That was reverse video characters which can now live in the EP text buffer too. The control characters also have their MSB set, but this is no problem for the printer handler to sort out.

The program works simply enough and should not hamper the typest. If the MODE SELECT key is pressed then the next character will be ored with 80H and put in the text. The next key pressed will be back to normal. If the MODE SELECT is pressed twice then all following characters will ored with 80H stored. This will continue until the MODE SELECT key is pressed a third time which caused the mode to return to normal. There are 3 keys which can not be stored in text in this program. They are MODE SELECT, LINE FEED and DELETE. I felt these should retain their usual fuction at all times. A small change in the program could modify this.

I had to get by 4 hrutles on this one. The first was to find a way to get a control character in in the first place. EP uses every control character available to perform immediate control action. So I needed some way to stop this. There is one key that EP does not use and that is MODE SELECT. So that part was solved - just write a routine that checks for MODE SELECT and then follow it with the control character or alphanumeric to be placed in text.

The next part was how to handle the control characters after they get to the VDM. Many of these have an effect on the display we don't want if our desire is to embed them in text for later use. I needed some way to "hide" them. This was easily handled with an ORI 80H instruction.

Then came the problem with the way EP handles the text buffer. Everytime something is moved in the buffer, which is practically constantly while inputting text, all the text gets shoved through a ANI 7FH shaped hole which unmasks our control character. As soon as this happens all sorts of things begin to change in the display! So I had to find these. There were five places where this occures.

The last was where to put the routine, considering that EP likes to clear all memory on start up. I could disable this, but I rather like it for several reasons. So I chose instead to make room by squeezing the message data for EP's command sub set. This gave me the bytes I needed plus an additional **36** for future use!

The following locations in version SS need to be changed to NOP's. Each address pair had been an ANI 7FH before.
261, 262; 4A2, 4A3; 4B4, 4B5; 4C1, 4C2; 8D3, 8D4.

Location 0090H had been CD 39 09 which called one of EP's keyboard input routines. Change this to CD 23 OD which calls the start of the new control character routine.

The following listing is the modified message data for the EP command sub set.

```
    OC82: 54 20 52 45 41 44 00 52 00            ; T READ   R
    OC8B: 54 20 53 41 56 45 00 57 00            ; T SAVE   W
    OC94: 57 4F 52 44 20 23 00 58 00            ; WORD #   X
    OC9D: 52 43 52 44 20 23 00 59 00            ; RCRD #   Y
    OCA6: 43 20 A0 41 46 54 00 43 41 41 00      ; C  AFT   CAA
    OCB1: 43 20 42 46 52 A0 00 43 41 42 00      ; C BFR    CAB
    OCBC: 43 4C 20 53 59 53 00 43 4C 52 00      ; CL SYS   CLR
    OCC7: 4A 53 54 49 46 59 00 4A 30 2D 31 00; JSTIFY   J0-1
    OCD3: 4C 4E 20 53 50 43 00 53 31 2D 35 00; LN SPC   S1-5
    OCDF: 50 47 20 53 50 43 00 41 32 2D 00      ; PG SPC   A2-
    OCEA: 50 47 20 4C 45 4E 00 47 2D 37 32 00; PG LEN   G-72
    OCF6: 4C 46 54 20 4D 52 00 4D 30 2D 00      ; LFT MR   MO-
    OD01: 50 41 47 45 20 23 00 4E 31 2D 00      ; PAGE #   N1-
    ODOC: 50 52 54 20 4C 4E 00 50 30 2D 00      ; PRT LN   PO-
    OD17: 4C 4E 45 20 4C 4E 00 4C 32 35 2D 00; LNE LN   L25-
    OD23: SPACE AVAILABLE FOR ROUTINES = 90 BYTES
    OD7D: START OF NEXT MESSAGE FILE            ; TIME
```

The source for the ConTroL INput (CTLIN) follows with an HEX dump. I haven't written a printer driver yet to make use of the added control features. We're just having fun with the inverse video. It's nice to know that the capability is now in EP for this and should come in handy for a Hytype driver I've got on the back burner. My regards to you, Stan, and to the members.

Sincerely,

H Leon Winter

H Leon Winter
Summer Institute of Linguistics
Nasuli, Malaybalay
Bukidnon, Philippines, 8201

```
ASSM OD23 5000

OD23                 0100 * CONTROL CHARACTER INPUT ROUTINE FOR ELECTRIC PENCIL
OD23                 0105 *
OD23                 0110 * ALLOWS CONTROL CHARACTERS TO BE "HIDDEN" IN TEXT
OD23                 0120 * ALLOWS REVERSE VIDEO CHARACTERS TO BE IN TEXT ALSO
OD23                 0130 *
OD23                 0140 * WRITTEN BY H. LEON WINTER           FEB 1982
OD23                 0150 *  AT NASULI, MALAYBALAY, BUKIDNON, PHILIPPINES, 8201
OD23                 0160 *
OD23                 0170 * EQUATE TABLE
OD23                 0180 KYBD2   EQU   0939H     ONE OF EP'S KEYBOARD ROUTINES
OD23                 0190 *
OD23                 0200 * IN ADDITION TO THE CHANGES NOTED IN THE DOCUMENTATION,
OD23                 0210 *  WHICH INCLUDE THE NEW COMMAND SUB SET MSG DATA AND THE
OD23                 0220 *  5 ANI 7FH'S THAT ARE REPLACED WITH NOP'S, THERE IS THE
OD23                 0230 *  INITIAL CALL TO THIS PROGRAM. THE LINE OF CODE AT 0090H
OD23                 0240 *  IN VER. SS OF EP MUST BE CHANGED FROM CD 39 09
OD23                 0250 *  TO CD 23 OD
OD23                 0260 *
OD23                 0270 CTLIN   ORG   OD23H     1ST FREE BYTE AFTER SUB SET MSG
OD23 CD 39 09        0280         CALL  KYBD2     CHARACTER USUALLY INPUT FROM HERE
OD26 79              0282         MOV   A,C       EXAMINE CHARACTER
OD27 FE 7F           0284         CPI   7FH       IS IT A DELETE KEY?
OD29 C8              0285         RZ    .         MAINTAIN DELETES FUNCTION
OD2A FE OA           0287         CPI   OAH       MIGHT BE A LINE FEED
OD2C C8              0288         RZ    .         IF SO, MAINTAIN IT ALSO
OD2D 3A 62 OD        0290         LDA   FLAG      SEE IF IN SPECIAL MODE OR NOT
OD30 B7              0300         ORA   A         IF FLAG IS 0 ALL IS NORMAL
OD31 C2 52 OD        0310         JNZ   MSSET     IS SET FOR SPECIAL MODE
OD34                 0320 *
OD34                 0330 * NOT IN SPECIAL MODE IF HERE
OD34                 0340 *
OD34 79              0350         MOV   A,C       EXAMINE CHAR THAT WAS INPUT
OD35 B7              0360         ORA   A         CHECK FOR MODE SELECT KEY
OD36 CA 3A OD        0370         JZ    MODE1     COMES THROUGH ANI 7FH IN KYBD2
OD39 C9              0380         RET   .         STILL NORMAL MODE, PUT CHAR TO CRT
OD3A                 0390 *
OD3A                 0400 * IF HERE THEN MODE SELECT KEY WAS INPUT FOR MODE CHANGE
OD3A                 0410 *
OD3A 3E 01           0420 MODE1   MVI   A,01H     SET UP FOR MODE CHANGE
OD3C 32 62 OD        0430         STA   FLAG      FLAG NOW REFLECTS CHANGE
OD3F CD 39 09        0440         CALL  KYBD2     NOW GET CHAR TO BE CHANGED
OD42 79              0450         MOV   A,C       CAME BACK IN C
OD43 B7              0460         ORA   A         0 IF ANOTHER MODE SELECT
```

CON'T ON PAGE 16

```
0D44 CA 4F 0D          JZ    MODE2     IF SO THEN NEED ANOTHER CHANGE
0D47              *
0D47              * IF HERE THEN JUST NEED TO INPUT 1 SPECIAL CHARACTER
0D47              *
0D47 F6 80             ORI   80H       REVERSE AND HIDE CHARACTER
0D49 4F                MOV   C,A       NEXT ROUTINE WILL LOOK IN C
0D4A AF                XRA   A         CLEAR A TO FIX FLAG
0D4B 32 62 0D          STA   FLAG      FLAG NOW SET FOR NORMAL MODE
0D4E C9                RET   .         NOW PUT CHAR TO CRT
0D4F              *
0D4F              * HERE BECAUSE CHANGE IS NOW SET TO THE SPECIAL MODE
0D4F              *
0D4F C3 23 0D     MODE2  JMP   CTLIN     BEGIN INPUT IN SPECIAL MODE
0D52              *
0D52              * SPECIAL MODE HOLDS UNTIL ANOTHER MODE SELECT KEY
0D52              *
0D52 79           MSSET  MOV   A,C       EXAMINE CHARACTER
0D53 B7                ORA   A         ZERO IF MODE KEY
0D54 CA 5B 0D          JZ    CLRMS     TO CLEAR SPECIAL MODE
0D57              *
0D57              * IF HERE THEN REMAIN IN SPECIAL MODE AND DO REVERSE VIDEO
0D57              *
0D57 F6 80             ORI   80H       ADD BIT FOR REVERSE
0D59 4F                MOV   C,A       SET UP FOR NEXT ROUTINE
0D5A C9                RET   .         PUT TO CRT
0D5B              *
0D5B              * HERE TO CLEAR THE SPECIAL MODE
0D5B              *
0D5B AF           CLRMS  XRA   A
0D5C 32 62 0D          STA   FLAG      FLAG NOW IN NORMAL STATUS
0D5F C3 23 0D          JMP   CTLIN     GO GET A CHAR FOR CRT
0D62              *
0D62              * THE FOLLOWING BYTE IS THE STORAGE FOR THE MODE STATUS
0D62              *
0D62 00           FLAG   DB    00        INITIALIZE IN THE MORMAL MODE
0D63              *
0D63              * THERE ARE 36D FREE BYTES REMAINING FROM THIS POINT
0D63              *
0D63              *
```

```
0D23  CD 39 09 79 FE 7F C8 FE 0A C8 3A 62 0D B7 C2 52
0D33  0D 79 B7 CA 3A 0D C9 3E 01 32 62 0D CD 39 09 79
0D43  B7 CA 4F 0D F6 80 4F AF 32 62 0D C9 C3 23 0D 79
0D53  B7 CA 5B 0D F6 80 4F C9 AF 32 62 0D C3 23 0D 00
```

ADAPTING ELECTRIC PENCIL TO SOL AT F000 HEX FROM C000
by Louis Bucklin

Enclosed is my check for renewal of Proteus. My thanks to you for your continuing efforts to bring us useful information about the Sol.

By way of contribution I am enclosing a listing of changes that I found necessary in order to adapt Electric Pencil to the Sol at F000 hex from C000 hex. It became painfully obvious rather quickly that unless one run's all software at F000 hex, the time and potential error of switching becomes too burdensome.

I made considerable effort to contact Michael Shrayer Software before attempting to track down the changes in the code that were required. But, letters were returned and phones proved to be disconnected. I found this unfortunate not only because searching for the changes was time consuming, but because it appears that all hope now for the upgrading of the Pencil has disappeared. While Pencil remains for me perhaps the single most effective piece of software written specifically for the Sol, it could benefit from error removal and extension.

In some ways the departure of the Electric Pencil is a little bit like the demise of the Sol--though I expect the reasons were quite different. It is an enormously valuable piece of software. There ought to be a means of keeping this alive. I am surprised that Shrayer didn't sell rights to the Pencil instead of allowing it simply to die. This would have been useful, especially for those who would like to further update the Sol to the larger screen. Perhaps there is someone who knows what has happened who could help on this. I suspect that this shift may be more complex than simply changing addresses.

In any event, the attached listing may be of some use to Proteus readers. The code addresses are relevant only to the Electric Pencil Ver SS-II for the teletypewrite, not for those which have the custom printer drivers. Hence, before changing any code, one should check version numbers. These are to found at addresses 110-15F hex. Also, since there may be several editions of this version one should also check to determine if the old values, as shown below, are in their proper locations. If not, they may be near by. For those who have other versions, my guess is that there would not be any other changes that those specified. Hence, it is only a matter of searching for the specified codes. Other than simply changing addresses in ram for the display, important changes were necessary in order to inform Pencil when the end of the display ram had been reached. This is the reason for the substitution of so many 00 for D0 codes.

Sincerely yours,

Louis P. Bucklin

March 7, 1982
1061 Via Alta
Lafayette, CA 94549

(Ed. note: I had an interesting conversation with someone who says he testified as an expert witness in a lawsuit between Michael Shrayer and the Vector Graphic computer company. It seems Shrayer tried suing Vector Graphic, claiming they had infringed on his copyright of Electric Pencil in their word processing software. The expert showed that large sections of Shrayer's code were identical to Processor Tech's code in the Software #1 editor, so Shrayer's case was shot down (he wasn't original author) and he skipped town to avoid counterclaims. That's why Electric Pencil disappeared from the marketplace, the story goes.---Stan.)

Code Changes to Adapt Electric Pencil to F000 on the Sol
Use for Version SS-II 12

Addr	OlCd	NwCd	Addr	OdCd	NwCd
171	36	3E	D6C	CC	FC
171	0D	00	EA3	CD	FD
17B	CD	FD	EDA	CC	FC
26A	CD	FD	EF0	CD	FD
28A	CC	FC	F1F	CC	FC
3CD	CF	FF	F39	CC	FC
3D6	CF	FF	F50	CC	FC
515	22	2D	F6A	CD	FD
516	2E	27	F81	CD	FD
644	CC	FC	F98	CD	FD
656	D0	00	FB3	CD	FD
820	CD	FD	10EA	CD	FD
970	CD	FD	1106	CC	FC
984	D0	00	1108	D0	00
A10	CD	FD	114D	CD	FD
A2D	D0	00	11D8	D0	00
A58	D0	00	11E2	C0	F0
A5E	CD	FD	120D	C0	F0
A62	CD	FD	1238	C0	F0
A65	CE	FE	128A	CD	FD
AC0	CD	FD	12DC	CE	FE
D3C	CC	FC	12FB	CE	FE
D43	CC	FC	1304	CE	FE
D69	CC	FC	130F	CE	FE

#
Addr = address in hex
OdCd = old code   NwCd = New Code

HELP NEEDED ON 1200 BAUD TAPES, G2 MICROSOFT AND McKELVEY' MCP
by Alastair Preston

I have several questions that readers of PROTEUS/NEWS may be able to answer for me. As background info, I am running an S100 system under CUTER (ROM) with a Z80 Cpu, 32K memory, 3P&S, CUTS and a VB1B video board set at C000H,

1) The CUTS board works fine at 300 Baud, and reads commercial 1200 baud tapes, but will not load "1200" baud tapes it has written. I suspect that the problem is in the 1200 baud write circuitry, but despite changing the relevant chips, the problem remains. Who knows how to fix this? Do you know of any reasonable repair shops in this area?

2) In view of all the discussion re moving SOLOS to F000, nothing has been said about CUTER-- are replacement ROMs available to put CUTER at F000 ?

3) I have both ECBasic and G2 Microsoft Basic; Has anyone implemented the ECBasic cassette routines in G2 ? (i.e. FILE#,PRINT#, READ#,REWIND# and CLOSE#). MICROSOFT does not have ths source code for G2 Basic and are unable to help.

4) In implementing M. McKelvey's "Micro Communication Package" (P/N v.3, #5/6), I have had trouble with the text transmission routines; they output everything including the line count bytes, line numbers and the non ASCII coding at the beginning of each line of a Basic listing, I was able to handle assembly listings and text produced with ALS8 and EDIT by reassembling the program with an extra INC H to skip one character at the beginning of each line , and editing out the line numbers in the recieving system's file. Any attempt to send a Basic listing bombs as the number of characters preceeding the line number is not constant. Has anyone solved this? Also, since I do not have a disk system yet, I would like to use cassetts instead. My attempte to make use of the rather limited information in the CUTS and CUTER manuals have been unsuccessful; I just have'nt figured out the proper procedure yet. Any suggestions?

5) I am assembling a disk controller board with hopes of being able to get some drives in the future if my economy improves. Is or will PTDOS be available on standard 5 1/4" disks? [Ed. note: Not in our plans.]

6) Does anyone know of an available Z80 assembler/editor (Zilog mnemonics) on CUTS tape; also a useable database and any Adventure games ?

Sincerely,

15 D Twin Terrace,
Edmonton, Alberta,
Canada, T6K 1V4
8 February 1982

Alastair Preston.

MORE ON MY GENERAL LEDGER PROGRAM
by Franz J. Hirner

Dear Stan,

Thank you for publishing the GENERAL LEDGER program. I had hoped that you would print the program listings but I imagine due to the length it was impossible. I recieved several requests for North*Star Single Density disk copies as well as Double Density. I am happy to provide either version to members for $25.00. While it is a lot of work, in view of the response I will also supply printed listings of the programs for $25.00 for those members who are not able to use the North*Star format.

Keep up the good work.

Sincerely,

Franz J. Hirner

631 Matsonia Drive
Foster City, California 94404
February 12, 1982

Franz J. Hirner

AN ALTERNATE 80X24 VIDEO UPGRADE
by Michael Carter

While the Micro Complex 80x24 upgrade (Vol 4, #5/6) is certainly impressive and a very welcome addition to the Sol repertoire, I would like to bring an alternative to the attention of your readers.

My partner and I are just completing the design of a 80x24 video board which offers several advantages over most alternative video boards including the Micro Complex product. By utilizing an on-board Z80 which is dedicated to handling the video function, it offers all the advantages of a memory-mapped display, without the overhead of address space or CPU time. All the necessary software will be contained in on board EPROM. In addition, it offers user programmable characters and limited high resolution graphics.Tentative specifications are:

80x24 display
Z80A processor, 4MHz
512 user programmable characters (128 defined on initialization, can be redefined)
4K EPROM
16K RAM
HI-RES graphics - 640×288 (Implemented with the programmable character generator)
Keyboard port - type-ahead buffer
Centronics printer port
Light pen interface
RF shielding on all high frequency components
Superscripts, subscripts, underline, strike-thru, blanking, half-intensity, flashing, inverse, etc.

The initial configuration is for an S100 video card. However, since I have two SOL's and regard them fondly, we will consider (if there is sufficient demand) producing a version specifically for the SOL, eg. as a piggy-backed board similar to the Micro-Complex . At the same time, we would arrange to disable SOLOS under software control (there would be no need for it, since the video driver routines would be located on the video board) and thus allow for a full 64K of RAM. We would also include a 4 MHz Z80 upgrade for the main processor.

I therefore invite interested readers to write to me at the above address, expressing their interest. We would expect to have the boards produced by approximately mid-year. We would be willing to sell bare boards, kits or fully assembled and tested units, and would expect the latter to sell at a price comparable with the Micro Complex board.

Regards

Michael Carter

4 Richardson St
Garran ACT 2605
AUSTRALIA

PS. For three years, I have puzzled over why SOLOS was located at C000 rather than (eg.) at F000. My partner has suggested a reason: C000 makes it simple to implement the PHANTOM with an Exclusive OR.

(Editor's note: Solos was located at C000 for compatibility with other products in the early Processor Tech line, namely ALS-8 firmware and VDM-1. When the biggest memory boards were 4K bytes, 48K seemed enormous and having Sol at C000 was no handicap.)

RELOCATE MDOS AND INTERFACE ECBASIC TO A MICROPOLIS DISK DRIVE

by Ron Shenk

The Problem

After adding a Micropolis Mod II disk drive to my SOL computer I had two BASIC interpreters, PT's ECBASIC which I had been using up to that time and the Micropolis disk basic which I call MBASIC. Both have nice features: MBASIC supports random access disk files, can achieve up to 60 decimal digits of accuracy if desired, and has nice program tracing. On the other hand, ECBASIC supports IF..THEN..ELSE, has a gem of an editor and most significantly is FAST. Typically MBASIC requires 30% more time to execute the same program.

How nice it would be if ECBASIC could be interfaced to the disk drive.

The Solution

Since both MDOS and ECBASIC occupy the same space in memory, one had to be moved. I had other applications which could benefit from a copy of MDOS at the top of memory, so I chose to move it. Two such benefits are: 1) one can use Micropolis' very fine DEBUGger to single step any CP/M .COM program, and 2) one can convert CP/M files into MDOS files and vice-versa.

With some effort I found all the addresses in MDOS, about 1600 in total, and wrote the program MDOSMOVER to increment them all by XX00H for any hex digits XX and move the code to its new location. I call this relocated disk operating system RDOS, mine occupies 92B1H to B800H.

Needed next was a disk executive for EXBASIC. SDEXEC sits above RDOS and requires 512 bytes of memory. Upon initialization SDEXEC modifies four I/O vectors in ECBASIC and some code in RDOS. At this point all input goes through SDEXEC but is completely transparant to ECBASIC. The executive takes charge however when it sees CONTROL D and prompts for a disk command. The commands are:

DOS (to exit ECBASIC and enter RDOS)
SOD "parameters" (to save a program on disk)
FEED "parameters" (to load a program from disk)
REPLACE "parameters" (to update a disk file)
and RET (to return to ECBASIC).

The resulting marriage between ECBASIC and SDEXEC I call SDBASIC.

SDBASIC works with source code on a character by character basis. When saving a program using SOD the executive decides to close the file when it receives END (blank,E-N-D,blank) or READY either from the keyboard or as the last statement of the program. By using source code the programs are transportable. A BASIC program written within the correct syntax of both BASIC's can be disk loaded and run on both (I also have rigged SDEXEC to work with MBASIC). Working with source code also has another advantage, programs LISTed while connected to a time-sharing service or other remote site and downloaded to disk can simply be fed and run. It is easy to modify SDEXEC to download source data arriving at the serial port.

For anyone interested, I have included a printout of SDEXEC. Further I can supply MDOSMOVER or SDEXEC on printout, on cassette tape, or on Micropolis Mod II diskette along with documentation. For a printout of MDOSMOVER please send $4. For a diskette or a tape, please send $15 for MDOSMOVER, $10 for SDEXEC or $20 for both.

Ron Shenk
329 Robin Hood Rd.
Atlanta, Ga. 30309

```
*
*                  SDEXEC
*
* A DISK DRIVER FOR ECBASIC (REV. A) ON MICROPOLIS
* USES MDOS ROUTINES AND IS COMPATIBLE WITH MDOS DISKETTES
* THIS AND MDOS MUST BE MOVED TO A LOCATION ABOVE BASIC
* CLOSES FILES UPON ENCOUNTERING READY OR  END  NOT
* WITHIN QUOTES.  INCLUDE END STATEMENT AS LAST LINE
* OF A BASIC PROGRAM.
*
*******************************************************
BASEADDR     EQU   2B00H
CREATE       EQU   0166CH
WTINXPOSI    EQU   018BAH
DOSADDR1     EQU   2022H
DOSADDR2     EQU   1C8FH
DOSADDR3     EQU   1C95H      ;DISKERR
DOSSTRTUP    EQU   1598H
DOSADDR4     EQU   204DH      ;COMTAB PTR
DOSADDR5     EQU   2053H      ;NO MATCH BYTE
MCOMTAB      EQU   20C0H      ;MICRO COM TABLE
DOSADDR6     EQU   2059H      ;NO MATCH JMP
INCRECPOS    EQU   015A0H
KSTAT        EQU   0C02EH
BUFSET       EQU   025B0H
OPENFILE     EQU   01709H
RFINXPOSI    EQU   0187AH
CLOSEFILE    EQU   01754H
DISKERROR    EQU   01C8FH
VDMOT        EQU   0C054H
NASCPAR      EQU   029ACH     ;29AC
DRIVENO      EQU   029D7H     ;29D7
INBUF        EQU   01A0H
FILEBUFFER0  EQU   0264BH
ASCIIBUFF    EQU   029E3H     ;29E3
UIPRT        EQU   0C800H
UOPRT        EQU   0C802H
IPORT        EQU   0C806H
OPORT        EQU   0C807H
ERRORMSG     EQU   01CB1H
SCRATCH      EQU   0163CH
@APP         EQU   2B00H      ;OFFSET FOR SDEXEC
*****************************************************
LAYDOWN      EQU   $
OFFSET       EQU   LAYDOWN-BASEADDR
             JMP   INTLZ-OFFSET
EXEC         NOP
EXEC1        DW    0
             CALL  KSTAT
             RZ
             CPI   04
             RNZ
             PUSH  H
ADR1         LXI   H,DPROC-OFFSET
ADR2         SHLD  EXEC1-OFFSET
             MVI   A,0C3H     ;JMP
ADR3         STA   EXEC-OFFSET
ADR4         LXI   H,GETLIN-OFFSET
ADR5         SHLD  BRNCH1-OFFSET
             POP   H
DPROC        PUSH  B
             PUSH  D
             PUSH  H
             LXI   H,0
             DAD   SP
ADR6         SHLD  SPFIX-OFFSET
BRNCH        JMP   GETLIN-OFFSET
BRNCH1       EQU   BRNCH+1
GETLIN       LXI   H,EPRMPT-OFFSET
ADR7         CALL  CILINE
```

CON'T ON PAGE 19

```
2240             DCR   B
2244 ADR8        CNZ   DOSADDR1
2248 ADR9        CC    DOSADDR2
2252 COMNOTFND   MVI   A,0BH     ;NOT FOUND CODE
2256 ADR10       CALL  ERRORMSG  ;OUTPUT ERROR MESSAGE
2260             MVI   A,0DH
2264 ADR11       JMP   RETN-OFFSET
2268 RETN        LHLD  SPFIX-OFFSET
2272             SPHL
2276             POP   H
2280             POP   D
2284             POP   B
2288             RET
2292 INTLZ       LXI   H,EXEC-OFFSET
2296             SHLD  0586H      ;IN SOL BASIC IN
2300             SHLD  267BH
2304             SHLD  2699H
2308             SHLD  26A4H
2312             SHLD  26F3H
2316 ADR12       LXI   H,DOSADDR3 ;DISKERR
2320             MVI   M,0C9H     ;SO RDOS RETURNS TO BASIC
2324 ADR13       LXI   H,COMTAB-OFFSET
2328 ADR14       SHLD  DOSADDR4
2332 ADR23       LXI   H,COMNOTFND-OFFSET
2336 ADR15       SHLD  DOSADDR5  ;IF NO MATCH
2340 ADR16       CALL  NULEXEC-OFFSET
2344             PCHL            ;TO SOL BASIC
2348 NULEXEC     LXI   H,0
2352 ADR17       SHLD  EXEC-OFFSET
2356 ADR18       SHLD  EXEC1-OFFSET
2360             RET
2364 COMTAB      DTH   'DOS'
2368 ADR19       DW    DOS-OFFSET
2372             DTH   'RET'
2376 ADR20       DW    INTLZ-OFFSET
2380             DTH   'FEED'
2384 ADR21       DW    FEED-OFFSET
2388             DTH   'SOD'
2392 ADR22       DW    SAVE-OFFSET
2396             DTH   'REPLACE'
2400 ADR24       DW    REPLACE-OFFSET
2404 * DTH 'BASIC'
2408 *ADR26 DW INTLZ-OFFSET
2412             DB    0          ;END OF TABLE
2416 DOS         LXI   H,DOSADDR3 ;DISK ERR
2420             MVI   M,0C3H
2424 ADR27       LXI   H,MCOMTAB ;COM TAB
2428 ADR28       SHLD  DOSADDR4
2432 ADR29       LXI   H,DOSADDR6
2436 ADR30       SHLD  DOSADDR5  ;IF NO MATCH
2440 ADR31       CALL  NULEXEC-OFFSET
2444             SHLD  IPORT      ;HERE AND OPORT TOO
2448 ADR32       JMP   DOSSTRTUP
2452 FEED        LXI   H,FMAIN-OFFSET
2456 ADR33       SHLD  BRNCH1-OFFSET
2460 ADR34       CALL  BUFSET     ;IN RDOS
2464 ADR35       CALL  OPENFILE   ;IN RDOS
2468 ADR36       JC    QUIT-OFFSET
2472             MVI   C,20H
2476 ADR37       JMP   GOBCK-OFFSET
2480 FMAIN       CALL  KSTAT
2484             CPI   80H
2488 ADR38       JZ    RESET-OFFSET
2492             MVI   B,0        ;FILENO
2496 ADR39       CALL  RFINXPOSI
2500 ADR40       JC    QUIT-OFFSET
2504 GOBCK       MOV   A,C
2508             CPI   0AH
2512 ADR41       JZ    FMAIN-OFFSET
2516 ADR42       JMP   RETN-OFFSET
```

```
2520 QUIT        PUSH  PSW
2524             CPI   02
2528 ADR43       JZ    QUIT1-OFFSET
2532             MOV   A,C
2536             CPI   13H
2540 ADR44       JZ    QUIT1-OFFSET
2544             POP   PSW
2548 ADR45       CALL  DISKERROR
2552             PUSH  PSW
2556 QUIT1       POP   PSW
2560 RESET       MVI   B,0          ;FILENO
2564 ADR46       CALL  CLOSEFILE
2568 ADR47       CALL  NULEXEC-OFFSET
2572             MVI   C,0DH
2576 ADR48       JMP   GOBCK-OFFSET
2580 SAVE        CALL  BUFSET       ;SETS DRIVE #, BUF ADDR
2584             MVI   B,0          ;FILENO
2588             MVI   D,08         ;FILE TYPE
2592 ADR49       CALL  CREATE
2596 ADR50       JC    QUIT-OFFSET
2600 ADR51       LXI   H,LIST-OFFSET
2604 ADR52       SHLD  LPRTAB-OFFSET
2608 ADR25       LXI   H,QTKWC-OFFSET
2612             MVI   M,02         ;2 KEYWORDS
2616 ADR53       LXI   H,INRTN-OFFSET
2620 ADR54       SHLD  BRNCH1-OFFSET
2624 ADR55       LXI   H,OTRTN-OFFSET
2628             SHLD  UOPRT
2632 ADR56       LXI   H,KW1-OFFSET
2636 ADR57       SHLD  KW1AD-OFFSET
2640 ADR58       SHLD  KW1PTB-OFFSET
2644 ADR59       LXI   H,KW2-OFFSET
2648 ADR60       SHLD  KW2AD-OFFSET
2652 ADR61       SHLD  KW2PTB-OFFSET
2656 INRTN       LHLD  LPRTAB-OFFSET
2660             MOV   A,M
2664             INX   H
2668 ADR62       SHLD  LPRTAB-OFFSET
2672             CPI   80H
2676 ADR63       JC    SDIT-OFFSET
2680             LXI   H,0
2684 ADR64       SHLD  EXEC-OFFSET
2688 ADR65       SHLD  EXEC1-OFFSET
2692             LXI   H,OPORT
2696             MVI   M,3
2700 SDIT        ANI   7FH
2704 ADR66       JMP   RETN-OFFSET
2708 OTRTN       PUSH  PSW
2712             PUSH  B
2716             PUSH  D
2720             PUSH  H
2724             LXI   H,0
2728             DAD   SP
2732 ADR67       SHLD  SPFIX-OFFSET
2736             MOV   A,B
2740             CPI   22H
2744 ADR68       LDA   QTKWC-OFFSET
2748 ADR69       JNZ   QUOTE-OFFSET
2752             XRI   80H
2756 ADR70       STA   QTKWC-OFFSET
2760 QUOTE       RLC                ;BIT 8 TO CARRY & 1
2764 ADR71       JC    DISK-OFFSET
2768             PUSH  PSW
2772 ADR72       LXI   H,KW1PTB-OFFSET
2776 ADR73       CALL  ADD-OFFSET
2780 NXTKW       DCX   H            ;HL PTS TO HIGH BYTE
2784             PUSH  H            ;OF CUR KW PTR TBL
2788             MOV   A,M
2792             DCX   H
2796             MOV   L,M
2800             MOV   H,A          ;HL PTS TO CUR KW CHAR
```

CON'T ON PAGE 20

```
2804            MOV   A,M
2808            SUB   B
2812            INX   H          ;PT TO NXT KW CHAR
2816 ADR74      JZ    SETCHR-OFFSET
2820            CPI   80H        ;LAST BYTE?
2824 ADR75      JZ    EXIT-OFFSET
2828            POP   D          ;RESET CHR PTR
2832            POP   PSW
2836            PUSH  PSW
2840            PUSH  D
2844 ADR76      LXI   H,KW1AD-OFFSET
2848 ADR77      CALL  ADD-OFFSET
2852            DCX   H
2856            MOV   A,M
2860            DCX   H
2864            MOV   L,M
2868            MOV   H,A
2872 SETCHR     POP   D          ;HL PTS TO NXT KW CHR TO CHECK
2876            MOV   A,H        ;STORE HL IN CUR KWPTB
2880            STAX  D          ;DE PTS TO HIGH BYTE
2884            DCX   D          ;OF CUR KWPTB ADDR
2888            MOV   A,L
2892            STAX  D
2896            XCHG             ;HL PTS TO NEW CUR KWPTB+1
2900            POP   PSW
2904            SUI   02
2908            PUSH  PSW
2912 ADR78      JNZ   NXTKW-OFFSET
2916            POP   PSW
2920 DISK       MOV   C,B
2924            MVI   B,0        ;FILE NO
2928 ADR79      CALL  WTINXPOSI
2932 ADR80      JC    ERR-OFFSET
2936 GBACK      LHLD  SPFIX-OFFSET
2940            SPHL
2944            POP   H
2948            POP   D
2952            POP   B
2956            POP   PSW
2960            JMP   VDMOT
2964 ADD        ADC   L
2968            MOV   L,A
2972            RAL
2976            ANI   01
2980            ADD   H
2984            MOV   H,A
2988            RET
2992 ERR        CALL  ERRORMSG
2996 EXIT       POP   H
3000            POP   PSW
3004            MOV   C,B
3008            LXI   H,OPORT
3012            XRA   A
3016            MOV   M,A
3020            MOV   B,A
3024 ADR81      CALL  WTINXPOSI
3028 ADR82      CALL  CLOSEFILE
3032 ADR83      JMP   GBACK-OFFSET
3036 REPLACE    CALL  BUFSET
3040 ADR84      CALL  SCRATCH
3044 ADR85      JMP   SAVE-OFFSET
3048 KW1        DT    ' END '
3052            DB    8DH
3056 KW2        DTH   'READY'
3060 LIST       DT    '    LIST'
3064            DB    8DH
3068 EPRMPT     DTH   'COMMAND?'
3072 KW1PTB     DW    0
3076 KW2PTB     DW    0
3080 KW1AD      DW    0
3084 KW2AD      DW    0
3088 LPRTAB     DW    0
3092 QTKWC      DW    0
3096 SPFIX      DW    0
3600 *********************************************
3602 SDEMODTBL  EQU   $
3604            DW    LAYDOWN+2,ADR1+2
3606            DW    ADR2+2,ADR3+2,ADR4+2,ADR5+2,ADR6+2
3608            DW    BRNCH+2,GETLIN+2
3610            DW    ADR7+2,ADR8+2,ADR9+2,ADR10+2,ADR11+2
3612            DW    RETN+2,INTLZ+2
3614            DW    ADR12+2,ADR13+2,ADR14+2,ADR15+2,ADR16+2,ADR17+2
3616            DW    ADR18+2,ADR19+1,ADR20+1,ADR21+1,ADR22+1
3618            DW    ADR23+2,ADR24+1,ADR25+2
3619 * DW ADR26+1
3620            DW    DOS+2,FEED+2
3622            DW    ADR27+2,ADR28+2,ADR29+2,ADR30+2,ADR31+2,ADR32+2
3624            DW    ADR33+2,ADR34+2,ADR35+2,ADR36+2,ADR37+2,ADR38+2
3626            DW    ADR39+2,ADR40+2,ADR41+2,ADR42+2,ADR43+2,ADR44+2
3628            DW    ADR45+2,ADR46+2,ADR47+2,ADR48+2
3630            DW    SAVE+2,INRTN+2,GBACK+2,ERR+2,REPLACE+2
3632            DW    ADR49+2,ADR50+2,ADR51+2,ADR52+2,ADR53+2,ADR54+2
3634            DW    ADR55+2,ADR56+2,ADR57+2,ADR58+2,ADR59+2,ADR60+2
3636            DW    ADR61+2,ADR62+2,ADR63+2,ADR64+2,ADR65+2,ADR66+2
3638            DW    ADR67+2,ADR68+2,ADR69+2,ADR70+2,ADR71+2,ADR72+2
3640            DW    ADR73+2,ADR74+2,ADR75+2,ADR76+2,ADR77+2,ADR78+2
3642            DW    ADR79+2,ADR80+2,ADR81+2,ADR82+2,ADR83+2,ADR84+2
3644            DW    ADR85+2
3646            DD    0
3648 ENDTBL2    EQU   $
```

THE AMATEUR'S GUIDE TO A 62K CP/M SYSTEM WITH MCSOLOS AND DISC JOCKEY 2D, VERSION B

by Philip N. Barnhart

April 4, 1982

I installed a Micro Complex 80 x 24 VDM in my trusty old SOL in December of 1981. I had been running a 48k CP/M system for a couple of years with Morrow Design's double density 8 inch controller, Disc Jockey 2d, version B. The standard version of the DJ2DB locates its bootstrap prom and ram from E000H to E7FFH. With the top of my useable memory at C000H because of solos anyway, E000H was well out of the way. Then along came MCSOLOS and the 80 x 24 VDM. Not only did I nearly double my screen size, great for my word processing and program development chores, but SOLOS also got moved to F000H. Now the limit on my useable memory was no longer the SOLOS module but instead the DJ2DB. I lived with a "mere" 56k CP/M system for three months until I developed a crazy idea. Maybe I could have a 62k CP/M system!

There was something in your article on the new big screen (in the Proteus News, Vol. 4, No. 5/6, p.1) about being able to "turn off" MCSOLOS and MCVDM "by outputting 1s to the two high order bits of port FC." That gave me my brilliant/crazy idea. Perhaps I could move the DJ2DB to F800H thus leaving room for a 62k CP/M system, and only turn on MCSOLOS and MCVDM when I wanted access from the keyboard or to the screen. This would leave me with a somewhat complex bootup procedure but would gain me 6k of space for my system.

I quickly called Morrow about a "Relocation Package" for the contoller on a Friday. On Tuesday I drove over and picked it up for $60.00 plus tax (your credit card number will get it shipped UPS also). The Relocation package consists of two chips and a rewrite of CP/M (They will want your CP/M serial number for that). I modified the new CP/M BIOS to turn MCSOLOS on and off when a keyboard or screen access was needed and proceeded to modify the disc controller board by pulling two old chips and installing the two new ones exactly as I had been told. It worked great on the first try, right?

CON'T ON PAGE 21

Wrong. When I turned the computer on the little lights on the front of both of the disc drives came on at once. I knew something was wrong. I turned it off and looked for smoke. Seeing none, I settled down to find out what was wrong. Stan, your hardware freaks already know the problem, but for those who know about a much (or as little) as I do, let me finish the story. It has a happy ending.

With a little thought I realized that when Micro Complex said they put MCSOLOS and MCVDM on an internal bus, separate from the S-100 bus, they did NOT mean that the signals from the two busses did not interfer with each other at the same address location. I assumed that MCSOLOS was interfering with the memory mapped I/O of the controller located at the same address. Sure enough, when I entered and executed a short program to turn off MCSOLOS, the drives settled down and behaved the way they are supposed to when the machine is turned on.

I realized that to make this overlap of addressing work I would have to configure the DJ2DB to be off when MCSOLOS was on and on when MCSOLOS was off. Is that possible? A long puzzled look at the DJ2DB documentation (always keep, but never read, the documentation for anything until you are stuck) suggested a way out. The DJ2DB can be configured to turn itself off at reset or on power up and to turn itself on or off depending upon the output on port 40H. Since I had never needed to turn the controller off before, it was set up to ignore port 40H and to be on all the time.

FIXING THE CONTROLLER BOARD: Three changes to the DJ2DB S-100 board are needed. (1) The jumper, J4, located between 2A and 3A at the top left of the board must be changed from A-B to A-C. Simply pull the little connector off of the top two pins and push it back on the bottom two pins. The middle pin is used either way. This change turns the board off at power on and at reset. (2) Connect a jumper from J3A to one of the 8 pads just above it between 11C and 12C. This enables the "bank select" feature through port 40H. I chose the pad labeled "7" because it required the shortest piece of wire and also would require the shortest modification to my BIOS as I will explain latter. (3) Turn paddle 7 of switch 1 to the "on" position (Switch 1 is located at 5D). This change inserts one wait state in bus cycles which read from the 2708 EPROM on the DJ2DB controller board. The Morrow documentation says this paddle must be off for 2MHz operation and must be on for faster than 2MHz operation. Why it is needed for this configuration with no change in clock speed I do not know, but on my SOL the DJ2DB with only changes (1) and (2) is "dead" and with all three it works fine.

SOFTWARE FIXES: The theory of this modification is that the system runs with MCSOLOS off all the time except when access to the keyboard or the screen is needed. The relocation package from Morrow comes with a relocated boot loader and system equates which expect to find the DJ2DB located at F800H. I had to modify the references to SOLOS to Fxxx from Cxxx to run with MCSOLOS at all. In addition I had to insert code like the following to turn off the DJ2DB and turn on MCSOLOS before each call to a SOLOS routine and just the opposite afterward.

Morrow's version of CP/M 2.2 for SOL comes with three special routines for input from SINP and output to SOUT. Changing those special routines and making one additional change in the warm boot procdure is all that is necessary, 43 bytes total. In COTTY, delete the jump to SOUT, and in its place insert the following:

```
        xra     a       ;turn off dj2db
        out     040h    ;by o/p 0 in bit 7
        out     0fch    ;turn on mcsolos by o/p 0 in bits 6 & 7
        call    sout    ;0f019h
        mvi     a,0c0h  ;turn off mcsolos by o/p 1 in
        out     0fch    ;bits 6 & 7 and turn on dj2db by
        out     040h    ;o/p 1 in bit 7
        ret
```

In CSTTY similar code should surround the call to sinp as follows:

```
CSTTY   xra     a       ;turn dj2db off and mcsolos on
        out     040h
        out     0fch
        call    sinp    ;get character from keyboard
        push    psw     ;save character
        mvi     a,0c0h  ;turn mcsolos off and dj2db on
        out     0fch
        out     040h
        pop     psw     ;retrieve character
(continue balance of CSTTY as written)
```

Change CITTY beginning at CONIN3 as follows:

```
CONIN3  out     040h    ;turn dj2db off and mcsolos on
        out     0fch
conin31 call    sinp    ;get a character from keyboard
        jz      conin31 ;wait until there is one
        push    psw     ;save character
        mvi     0c0h    ;turn mcsolos off and dj2db on
        out     0fch
        out     040h
        pop     psw     ;retrieve character
        ret
```

In addition, inserting the following code at "wboot" will allow a direct warm boot from MCSOLOS. This is handy if you need to use a reset for any reason and the resident CP/M system is intact.

```
wboot   mvi     a,0c0h  ;turn mcsolos off and dj2db on
        out     0fch
        out     040h
(continue with existing code)
```

The only other issue is how to get from MCSOLOS at F000H to DJ2DB at F800H to cold boot the CP/M system. I have solved it with a somewhat inconvenient kludge. There is plenty of room in the MCSOLOS prom to allow the "load" key to download the short program described below and then jump to it. I just haven't gotten around to having a new prom burned yet. I cold boot as follows: First, in MCSOLOS, type, EH 0<cr>. This allows you to enter hex code directly into memory at 0. Then enter the following 9 bytes: 3E C0 D3 FC D3 40 C3 00 F8, return to command mode and enter EX 0<cr> or WB<cr>. If everything is right the 9 byte program will turn off MCSOLOS, turn on DJ2DB, and jump to DJ2DB's cold boot routine. Thereafter, the switching of MCSOLOS and DJ2DB will be transparent to the user.

For the Relocation Package write or call Morrow Designs, 5221 Central Avenue, Richmond, Ca. 94804, (415)524-2104.

Stan, keep up the great work. Without Proteus News to get the word out we would not know about the great new products for the SOL from Micro Complex and I would still be running 48k and a little screen.

Sincerely yours,

Philip N. Barnhart

Philip N. Barnhart
1709 Rose Street
Berkeley, Ca 94703

TELECOMMUNICATIONS SYSTEM FOR THE SOL/HELIOS II COMPUTER

by Allen Fincher

This program consists of three files that comprise a telecommunications system using the SOL/Helios II as a smart terminal. They have been submitted to Proteus for inclusion in the Helios library.

The program was written mainly for communicating with THE SOURCE, a time-sharing information utility located in McLean, Virginia (Source Telecomputing Corp., 1616 Anderson Road, McLean, VA 22102). It will automatically log you onto THE SOURCE via either TELENET or TYMNET, two telephone carrier companies. The auto-log feature checks the carrier detect signal from a modem to determine when to start the log-on sequence. While used primarily for THE SOURCE, provisions are included for manually logging onto any time-share system by selecting that option when the menu is displayed, although there may be conflicts between my program control code outputs and a non-SOURCE network (listed below).

While the program listing is copyrighted, I grant the original purchaser of the disk the right to make copies for his own use, and to assemble, use, and modify as desired so long as it is not sold or used commercially. I ask that you please honor this request. A lot of time, and connect-time expense, was involved in writing it.

Besides logging you automatically onto THE SOURCE, you can also upload data to, and download data from, the time-share computer.

The origional version of this program is written in Z-80 on my S-100 computer. Since I don't have a SOL, there is a remote possibility that corrections will be necessary to make the program run properly. I've tried to provide a program that can be assembled, loaded, and run without anything more than having you include some data in a few places in the source listing.

Because I wrote my current monitor by combining CUTER and a Z-80 monitor, I believe that I understand the necessary protocol for I/O operations via SOLOS (I also had the source listing of SOLOS Version 1.3, Release 77-03-27 to go by). I would like to gratefully acknowledge Fr. Thomas McGahee for his Proteus articles on the SOLOS output routines which supplied the serial status port bit definitions. Also, he noted that the SOL serial port is wired as a terminal, not a computer (see Proteus vol. 3, no. 4). This means that a modem can be connected directly to the SOL serial port without crossing any of the wiring between them. This program, with the terminal keyboard read routine modified for my system, has been tested on-line with THE SOURCE.

Here are the instructions for assembling the program:

1. If you use either TELENET or TYMNET as your telephone network, you will have to set the conditional assembly flags that are near the end of the TELCOM:S file. Set the appropriate EQUate to 1 and the other EQUate to 0. If you have access to both networks, then assemble this program twice equating first one of the networks, and then the other using two object files. I named by two files TELENET and TYMNET for simplicity. Also, make the data at SYSNAM in the Change Password routine in file TLCM2:S match the name of your object file.

2. Make any changes to the ONLINE routine necessary to make it work with your modem. It is supposed to wait until the modem has locked onto the time-share system carrier signal.

3. If you are a subscriber to THE SOURCE, enter the necessary data to the LOGON data statements near the end of TLCM2:S. I would suggest that you DO NOT include your password in the source listing. Instead, change it in the disk object file using the Change Password command when the menu is displayed. Because of "filter" routines, the password cannot contain control characters.

4. Assemble TELCOM:S, then perform an EXTRACT command to consolidate the object code into one record as follows:

   EXTRACT objectfilename,S

   If this is not done, the Change Password command will not be able to find the data in the disk object file.

The I/O of your SOL does not have to be defined (using I= or O=) before running the program. The terminal output is sent to an internal VDM driver and the SOL keyoard is used as the terminal keyboard. The modem I/O routines dynamically change the pseudo port by calling the AINP and AOUT vectors in SOLOS.

Because most time-share systems expect your terminal to have an 80-column display, I wrote the internal VDM driver to move any word that may be cut in two onto the next line. Most time-share systems also expect the terminal to have 24 lines, so to prevent text from scrolling off the top of the screen before you can read it all, press CTRL-S when desired to stop the system from transmitting, then press CTRL-Q to cause it to resume.

The menu options are:

1: Connect to The Source
2: Manual log-on
3: Change Source password
4: Quit & return to PTDOS

Option 1 automatically logs you onto THE SOURCE if you have included the necessary information in the data statements as outlined above.

Option 2 skips the auto-log routines and puts you directly into the main program I/O loop (terminal simulator).

Option 3 allows you to change your SOURCE password that is in the disk object file of the program your are running. THE NAME OF THE OBJECT FILE FOR THE PROGRAM YOU ARE RUNNING MUST HAVE BEEN ENTERED AT 'SYSNAM' IN THE SOURCE LISTING.

Option 4 will return you to PTDOS. You should sign-off of the time-share system before doing this.

The commands available are SEND, SAVE, and CATalog. They are entered by first typing a left-brace "{" and then the command.

Example: {SAVE filename/u

will save all incoming text in file 'filename' on unit 'u' until you enter "}". All incoming text is displayed on the VDM also. Anything you type between the "{" and carriage return is NOT sent to the modem, neither are the brace or the carriage return; they are strictly local. These commands can be used at any time that there is no incoming text.

CON'T ON PAGE 23

The SAVE command saves all incoming text in the file 'filename' as noted above. If the file does not exist, it will be created with type T, and with a block size of 4C0. All text is saved in a buffer. When 24 lines have been received, a CTRL-S is automatically sent to the network to stop the flow, and after all residual characters have been received, the buffer will be transfered to PTDOS. Since PTDOS also maintains a buffer, the transfer may not always result in an actual disk write taking place. After the transfer has taken place, a CTRL-Q is automatically sent to the network to tell it to resume sending.

The SEND command sends a file to the time-share system. When the end of the file is reached, the file will be closed (as shown by "}" being displayed) and the terminal bell (if you have one) will beginning ringing at about a 1 second rate and the message "(Press MODE to continue.)" will be displayed. Because this program has "filter" routines to prevent sending or receiving most control characters, object files cannot be handled. The actual through-put will seem slow (about 10 to 15 characters/second). This is caused by the program looking for an echoed character before sending the next one so that if the network pauses to service another customer, none of your data will be lost.

The CAT command will display the filenames of all type T files on unit 1 on the screen. They are not arranged alphabetically. When this command is used, a CTRL-S is automatically sent to the network, as is a CTRL-Q after listing the file names.

The following keys/control codes perform special local or remote functions:

CTRL-F: returns you to the menu.

The MODE key is used to abort commands, with the exception of the CAT command (PTDOS apparently ignores it as long as you are not in PTDOS). It is also used to abort the auto-log feature in case of problems; it skips to the main terminal loop so that you can manually log on to the system. If pressed during SAVE or SEND operations, the files will be closed and control will be returned to the main loop. The MODE character is NOT transmitted.

The DEL key may be used as back-space. It is changed to a CTRL-H (BS) when typed.

ESCape: used by THE SOURCE for sending electronic mail.

CTRL-E: used by THE SOURCE for identifying the ID of the person to whom you are chatting.

CTRL-G: sends a bell character.

CTRL-H: used by THE SOURCE as back space.

CTRL-P: used by THE SOURCE to terminate/abort an operation.

CTRL-Q: tells the telephone carrier network to resume sending after having sent a CTRL-S.

CTRL-S: tells the telephone carrier network to stop sending (pause).

(Editor's note: The source files will appear in the next Helios H-disk issued. We're waiting to fill it a little more.)

HARDWARE INTERRUPTS FOR THE SOL CON'T FROM PAGE 4

this feature can be added to the Sol by providing a vectored interrupt chip and associated hardware.

The Intel 8214 is a priority interrupt control unit. It has 8 interrupt request pins, coming from 8 interrupting devices (such as the keyboard, serial port, etc.) and one interrupt output pin that goes to the 8080's interrupt input pin. The 8214 monitors the 8 inputs and if any want service, it signals the 8080 and puts the RESTART instruction corresponding to the request into the 8080 via the data bus. The RESTART comes in 8 types (RST 0 through RST 7) which act like one-byte call instructions to predetermined addresses in the beginning of memory. If several requests come in at the same time, the 8214 can select the one with the highest priority first (determined by which pin it is assigned to). The processor can control the action of the 8214 in a limited way.

The Intel 8259A is a more sophisticated interrupt controller that can be programmed by output instructions directed to it through output port addresses. It works in a similar fashion to the 8214, but outputs a CALL instruction and the 16 bit address of the interrupt service routine. So it allows the interrupt service routines to begin anywhere in memory. The way it determines the priority of interrupts can be altered in many more ways than the 8214.

Several interrupt controllers can be cascaded together to allow more than 8 levels, but usually 8 is all that is needed.

So to get Leon Winter's Hytype printer to signal the 8080 when it is ready for another character, Leon will have to add an interrupt controller and hook up the printer's ready signal to one of its request pins (perhaps through an inverter if the signal has the opposite meaning than the controller wants).

He also has to provide an interrupt service routine that will know how to get more data to send to the printer. This is usually done through an area of RAM set aside as a buffer. His computing program puts bytes into the buffer and enables the printer to interrupt the computer. Whenever the interrupt occurs, the service routine checks to see if data is waiting in the buffer, and if so sends out another character to the printer.

This clears the ready signal (printer is busy) so the interrupt request is turned off and the service routine can return control to the main computing program where it left off. When the buffer is empty, the printer interrupt has to be disabled, so that the computer is not constantly being interrupted by the idle printer. The 8259A gives complete control in designating which interrupting devices are enabled or disabled. The 8214 allows a priority level to be set by the program, disallowing lower priority devices to interrupt the system.

Priority interrupt S-100 boards used to be made, but I haven't seen them for a long time, probably because most interrupt driven systems put the chip on the CPU board now. Anyway, it is a waste of a slot to do it that way. We should do it by piggybacking onto the Sol.

The 8214 requires four bits of one output port as its control port (to let the 8080's program set the priority threshhold). It should be possible to squeeze this out of the Sol's address decoder. However, the 8259A requires more ports for complete implementation, so it will be trickier to install.

In any case, the Sol has to be modified by installation of jumpers as described in the PTC update for vectored interrupt. These will prevent the onboard address decoder from becoming confused by the interrupt instruction that is placed onto the data bus.

If anyone has figured out just how and where to put the 8214 or 8259A onto a Sol, please send us the details. The whole story will appear in the Encyclopedia Processor Technica and as much as possible will go into Proteus News.

KEYBOARD IN THE TROPICS
BY H. Leon Winter

I have just struggled through 3 days of keyboard troubles on my Sol-20. After I finally figured out, and fixed the problem, I thought that others might run into this and would be interested in the solution. If I've missed seeing information on this published in past Proteus News and this is not a new account of the problem and the cure, Stan, please just file this letter.

My Sol lives in the tropics so this trouble may turn up sooner here than in the States, but it still can and may have happened to some of the members. One symptom was an unstable keyboard such that certain keys would not work at times. The most unsettling symptom was the generation of characters without anyone touching the keys! I could turn on the Sol and it would add 2 or 3 chatacters here and there. Or I could leave it, then come back later and find the screen full of garbage. If this happens to you, don't turn the Sol off. Grab a pencil and paper and jot down some of what you see. This will help solve the mystery.

After I scratched my head, I looked up the keyboard section in the Sol systems manual and reread it. This is always a good practice when any trouble is incountered. The time spent reading will more then pay you back in shortened trouble shooting time!

I made an assumption that the characters that "randomly" appeared on the screen had significance. Ultimatly nothing is random in digital logic. I was rewarded as I studied schematic X-22. The common factor appeared in the key matrix part of the drawing. In my case, it was pin 4 of the analogue multiplexer, U-22 which is a 4051. All of the screen garbage could be traced to this line including the fact that every time I'd touch REPEAT the computer would reset itself. (I had changed my keyboard some 2 years before so the REPEAT and BREAK were the resetting keys. It bugged me to have to be always retoggling the ALPHA SHIFT key after every reset! The BREAK key is also on pin 4 of U-22 hense the reset.)

The next thing was to grab my trusty logic probe. I have the type of probe that can be switched between either TTL or CMOS logic families. As it turned out, this feature was a help that I would not have known before doing this particular fix. Going along the input pins of U-22 with the prob in the TTL position, I could find no diference, all the lines read high as they should if all is well. But all was not well so I though about the fact that this is high Z circuitry and switched the prob to CMOS. Now the fault became quite evident. All the other lines still had a brite red high LED on the probe, but the offending line was dim.

The input lines on both U-22 and U-19 are pulled high by 33K ohm resisters. So something must be acting as a low impedance on pin 4 for the level to read as it did. Carefully, I removed all the screws holding the key assembly to the circuit board. Then I used the "end of my nose", as technicians call it, and traced the line from pin 4 to each of the keypads it services. On my way along this line I came upon 2 places where there was a patch of corosion across several closely orientated traces. (Living in the tropics, this type of thing does not surprise me as we have a high humidity. I think that living in a place where there was salt air could do this too or just age.) When I cleaned these, the keyboard troubles disappeared. The cleaning device I used was a fiberglass circuit board cleaning brush. This does a very good job of cleaning boards and is not as destructive as a little wire brush would be.

If you have the board apart for this or any other reason, take advantage of the siturtion and do a good cleaning of the keypads on the circuit board. I use clean gauze and rubbing alchol. Then dry the surface off with another clean gauze. Carefully reassemble the board. When all the screws are started, tighten them evenly but not too tight. Remember: this is not the head casting on your Chevy engine, it's very thin plastic! There is not much loading on the board and there are so many screws that lightly is more than tight enough.

There, Stan, is the tale of trouble found. I always like happy endings, hi. Thank you for putting out Proteus. I get a lot from what others contribute, as I'm sure you do too.

Sincerely,

H Leon Winter

H Leon Winter
Summer Institute of Linguistics
Nasuli, Malaybalay
Bukidnon, Philippines, 8201

LETTERS TO THE EDITOR:

...in Response to Bruce Diller's Keyboard Fix

Seasons Greetings to you and yours. I hope this letter finds you in good health, and ready to start the new year.

For Christmas, I bought my SOL a new printer that has true descenders, and full width paper capabilities. Since my job requires me to write reports and tabulate information, both these features are useful for me, while others might not find it so.

I have been meaning to write in response to some articles in your last issue. I have had 3 failures on my 16KRA board, and in each case I have been able to plug in a new IC and get it back on the line. One failure was in the refresh counter, one was with the Page Multiplexer, and one was one of the memory address drivers. Of the three, the last mentioned one was the hardest to fix, as the 75365 chip is as rare as hen's teeth. I was able to locate some for a $50 minimum order. I now have some of every chip on the board. As Joe says, eventually the delay line may go, and then it will be the end of it. My last memory acquisition was a Godbout "RAM XX" board populated to 24K. It is a static memory board, and it seems to me that it generates less heat than either of my 2 dynamic memory boards. In reponse to Bruce Diller and his keyboard fix, while his suggestion may work, it may also be weakening his circuitry. The original pads are backed with mylar, not aluminum foil, and it is non-conductive. On drawing X-25 of the Sol Manual, it indicates that the signals are capacitively coupled, (even your fingertip will make the circuit work) so by shorting the circuitry repeatedly, one may be encouraging component failure. I have been fighting key drop-out since day 1,and my latest experiment has been to loosen the screws that hold the keyboard to the keyboard printed circuit board. I stumbled on this by accident. After dismantling the keyboard for its annual cleaning, I had the usual problems getting all the keys to work. When I loosened the screws to shift alignment of the board, all the keys worked, so I simply tightened the screws enough to keep the big chunks of dirt out. Perhaps board distortion is the problem, and not dirt.

I anxiously await the next issue of PROTEUS NEWS, which I know will be out soon, as it is membership renewal time.

As always,

Jeff Tom

Jeff Tom

(Editor's note: The keyboard customizer kit in the new Proteus catalog includes extra foam contact pads from Keytronics, the keyboard maker. When your keys die, now you can fix them the right way. Personally, I never had one go bad, but if one did it would be costly to repair -- Keytronics has a $50 minimum order. That's why I include a few pads in the kit.)

...On McDPM, N*Disk, The Last Memory

Dear Stan:

I just received Vol. 4 #5/6 of PROTEUS NEWS with the renewal notice. A check to cover my subscription for 1982 is enclosed. The coming year would feel pretty empty without the regular delivery of the NEWS.

I have made some rather significant changes related to my SOL-20 this past year. I acquired and installed Micro Complex's Dual Personality Module. Not in its "standard" configuration, but with SOL's dedicated address space in the range E000 - EFFF. This required a rewrite of Micro Complex's SOLOS MONITOR and a revision of the wiring changes on the PCB. My reason for a departure from what might appear the idle was to avoid the exorbitant price North Star quoted for an alternate set of PROMS for my single density disk system. I was never happy with North Star's choice of the range E800 - EBFF for the single density disk controller. While the price was still the reasonable $25, I purchased a set of PROMS for the range FC00 - FFFF and used the range F000 - FBFF for an exceptional useful debugging routine (disassembler, single or multiple step tracer, string searcher, and program relocater) held on 2708 EPROMS. That left the range E000 - EFFF the next best choice for the new Personality Module.

I have also been one who has parted company with the rest of the North Star users by placing the disk operating system (DOS) on EPROMS in the address range beginning with D000. This required disassembling the standard DOS and BOOT and rewriting the entire system in assembly language compatible with the INTEL 8080. The pay-off was relocatability and the addition of exit commands to SOLOS, to the debugger, to NS BASIC's warm start (0004), and to CP/M. Oh yes, my version of DOS performs the unit disk management functions, namely, zeroing a specified disk drive, reading a specified sector from a specified drive and writing a specified sector from a specified drive, all of which makes for simplicity in the design of BIOS in my version of CP/M for the North Star disk system. Incidently, I do not intend to use the Micro Complex PM for switching SOLOS between the locations F000 and C000. I much prefer calling the "old" SOLOS into RAM at C000 from disk when I need to perform any operations with tape. In fact, I have removed the 2716 containing the old version from Micro's module to reduce unnecessary current drain from the PCB voltage regulator. A future project will be to extract the tape routines from old SOLOS and tuck them into some space I have available on EPROMS above D000.

The second rather extensive modification of the total system was the replacement of the LA-36 DEC printer-keyboard by the Microline 83A Printer. Since I have frequent occasions to let my SOL talk to an AMDAHL 470V/8 via a 300 baud modem, some fix was necessary in order to avoid taking the metal raps off of the SOL each time a change from 1200 baud to 300 baud was made. Of course, there was no good justification for replacing the LA-36 if the higher speed (120 cps) of the Microline printer was not readily available. The solution was to bring three leads out to a single-pole double-throw (center neutral) switch on the wall via three unused pins on jack J1 of the serial port. This joined my panel of switches for either routing the transmission of data to the AMDAHL or to the Microline printer. At the same time, software changes to NS BASIC and my assembler in order to take advantage of the top of form (TOP) function available on the Microline were suggested. Now the jump to the top of the next page after printing 62 lines is automatic.

My very next project will be to organize and install THE LAST MEMORY board which I have on order from Static Memory Systems in Freeport, Ill. This is the board that uses the new 2016 byte-wide 16K static RAM's which are replacable with 2716 EPROM's. The organization of the board will be as follows:

0000 - CFFF	RAM with 26 on-board 2016 memory chips
D000 - DFFF	ROM with 2 on-board 2716 EPROM's
E000 - EFFF	2 empty sockets. Space for SOLOS PM
F000 - FBFF	ROM with 2 on-board 2716 EPROM's (see note)
FC00 - FFFF	0 empty socket. Space for NS controller (see note)

Note: The last two address ranges are in conflict with the 2K boundaries between sockets. By design of the board, an empty socket occupies no address space. Actually there is no conflict with the address space covered because the FF HEX bytes read from the selected empty socket disable the read data buffer. Since the North Star controller requires only 1K, and I can find good use for the 3K from F000 to FBFF for debugging routines, I have proposed to Static Mememory Systems that I fill the 1K half (FC00-FFFF) of the last 2716 that conflicts with the controller with FF's. They agreed their "FFH detector" (a 74LS30 IC) would not discriminate between these FF's as apposed to those coming from an empty socket, would turn off the read data buffer and would therefore avoid any conflict with the overlapping disk controller. I believe that they were very pleased with the added flexibility which their board is found to have.

I was a little worried what my next project would be until the PROTEUS NEWS arrived. But now I can look forward to a 24X80 screen! Let us know all the details on your own experience in the next issue.

Thanks again for your thoughts and the news from others. If you feel that this rambling letter is of interest to others, don't hesitate to include it in the NEWS.

Regards,

Bob

P.O.Box 2240
Ann Arbor, MI 48106
January 22, 1982

Robert C. F. Bartels

Dear Stan:-

This is a postscript to my letter of Jan. 22, 1982. The Last Memory from Static Memory Systems arrived Feb. 9, 1982, has been installed in my SOL and works beautifully. The backplane board now contains only the North Star controller and the Last Memory. As explained in my last letter, the higher address half of a 2716 EPROM overlaps the address space into which the N* controller is mapped. There is, however, absolutely no conflict between these two since the half of the 2716 which maps into the address space common to both is filled with FFH bytes.

I repeat, the Last Memory performs exceptionally well and the power consumption has been reduced considerably. I have only one fault to find with the board. I am sure that my complaint would not be shared by many. I would have preferred that Static Memory Systems used "port addressing" for bank selection as does Godbout on the XIII memory board rather than their method of decoding the eight (8) IEEE-696.1 extended address lines on the S-100 bus. These lines were not defined when SOL was conceived. I have been using the port addressing to turn off the portion (16K) of my Godbout board which overlaps a SSM Microcomputer Products EPROM programmer. Thus, when burning an EPROM, I would plug the programmer board into the alternate S-100 socket on the top of the backplane and my burn program would send the disabling byte to memory via the 8080 OUT instruction prior to beginning the burn sequence. I can, of course, remove the Last Memory board and replace it with the old complement of RAM and ROM boards when the occasions arise to burn other EPROM's. But I am toying with the idea of using a VECTOR plug board to hold the bank select circuitry for disabling/enabling a segment of the Last Memory board via one of the unused extended memory lines on the S-100 bus. Don't yet know how to do it, but that's the fun of having a personnel computer not made untouchable by IBM.

CON'T ON PAGE 26

The plan mentioned in the last letter for extracting parts of the tape routines from SOLOS has been implemented. The commands SAVE, LOAD, YANK and CAT are now contained in a 4K utility subsystem that was described in the last letter as my "debugging routine" and which contained a disassembler and other useful programming tools. This subsystem resides on EPROM and is callable from the SOL Monitor with the command UT. When in the utility subsystem, the command

LOAD name (addr)

for getting a named file from tape and loading it as specified in the header or at the optional address (addr) is the same as with GET under SOLOS. The name LOAD for this command rather than GET was chosen to maintain consistency with the syntax used in my North Star BASIC where a file is pulled from the disk also with a LOAD. The utility subsystem command

YANK name (addr)

performs the same task as the LOAD with the exception that the tape CRC READ errors are ignored. Commands SAVE and CAT perform the same tasks as do those with the same names in SOLOS.

As one who prefers the use of disks and who uses tapes only infrequently, and then mostly for backing up the system programs, I have only one tape drive. Also, I have never used any tape speed other than 1200 baud. Hence my versions of the tape routines in the utility subsystem will operate only with tape drive #1 and at 1200 baud. The purpose of this was program space economy.

Again, if you think that the previous letter and its postscript are of interest to the members, include them -- or parts thereof -- in the NEWS.

Yours truly,

Bob

P.O.Box 2240
Ann Arbor, MI 48106

Robert C. F. Bartels

...Finding a Z80 Dissaassembler

It recently occured to me, that with the introduction of the Z80 upgrade for the SOL forthcoming from Micro Complex, there could be some possibility of converting Z80 programs available for other machines for use on the SOL/Z80, and for those like myself who are running Z80/System B or similar machines under CUTER. Graphics programs may be a problem for the SOL and VDM1 users, but could be used by those using video boards with graphics capability. (SOL users could forego the main video circuits and use an S100 graphics board such as the SSM VB1.)

I have for some time been trying to get the Sorcerer version of "Adventure" running on my system -- their cassette format is virtually identical to, and apparently copied from, CUTS. The only differences are the sixth character of the header (55h rather than 00), and the block length (one extraneous character extra at the end of a block, for CRC checks). My only troubles have been in (not) deciphering the direct video output routines. A good Z80 dissassembler would help, but I can't find one. Perhaps some liasion with the SORCERER users group could produce some results - such as a short interface routines to enable cross - loading of CUTS and Sorcerer tapes. We would benefit by access to their Z80 software and they could make use of the Proteus Cassette Library, particularly the Disk- tape programs. Hope these ideas don't sound too far-fetched! Oh, and don't forget all that TRS-80'system' software waiting to be converted!

Yours,

Alastair Preston

15 D Twin Terrace,
Edmonton, Alberta,
CANADA, T6K 1V4
22 February 1982

...Key Repeating and Underlining

Stan,

I have enclosed my subscription renewal form, along with a check in the amount of $24 to cover the cost. I really value this newsletter as I'm sure others do- and we all appreciate your effort in all the responsibilities associated with PROTEUS. If there is anything I can do to lighten the load, please let me know.

I appreciated all the good words and reviews regarding Bob Hogg's 80x24 screen mod. I haven't purchased it yet, but it probably is just a matter of time(and money). I also think the Z-80 mod would be a real asset for our SOL machines.

I am still using a single density N* disk controller, and will probably upgrade to Bob Hogg's new double density N* compatible controller board. However I would like to hear from other users who have purchased it, and find out how they like it.

A couple of months ago I implemented the 'key' repeat hardware mod(submitted by Jack Kinney, Vol. 4 no. 1), and it works great, it eliminates the need to hold down the repeat key - a nice little useful feature.

Another hardware mod I implemented was changing the solid cursor to an underline. I never did like that BIG white rectangular cursor, so I opted for the underline. This modification was written up in Dr. Dobbs Journal (March 81), specifically for the VDM board. I wrote the author regarding the mod for the SOL, and he sent me a detailed diagram - I will try to get his permission to pass it along to PROTEUS for other readers to try if they wish. It works well, however with some of PT's early game software that used the solid cursor you don't get the same affect. Although I haven't tried it - a switch could be installed to select either solid or underline cursor.

Another computer related goal this year will be to get CP/M up and running under N* DOS. I met another SOL owner in town the other night at a CP/M users group meeting and he has version 2.2 up. My reluctance so far has been the fact that I do not have a DD Dual disk system yet. I seems to me that CP/M needs a DD Dual disk system to be really effective.

Best Regards,

Rick
Rick Downs

RICHARD E. DOWNS JR.
P. O. BOX 440357 • AURORA, COLORADO 80044-0357

UNCLASSIFIED ADS

FOR SALE SOL 20 8K of static RAM. B&W Panasonic monitor available. SOLUS personality module. Call (415) 948-5832 evenings for more information. Richard F. Otte

ALS-8 assembly language development system in ROM for Sol--$75. Extremely fast 8080 assembler, editor, simulator on 8 K EPROM board by Processor Tech. Software documentation in Proteus's Encyclopedia Processor Technica volume 5.

HELIOS disk controller/formatter board set for sale -- $200. In excellent, error-free condition. This is a spare board set we no longer need. Use it as a back-up in case yours goes out or to swap-in as a diagnostic tool. Software and documentation available.

Contact Proteus, 1690 Woodside Road, #219, Redwood City, CA 94061, (415) 368-2300.

FOR SALE: North Star SA-400 Disk Drive (for SOL) including cabinet, power supply, S-100 controller board (SD) and cable. Also a 16KRa Memory board. $350.00 or best offer. Contact: Terry Walters, 441 Swallow Ct. Livermore, CA 94550 (415) 449-0536.

FOR SALE-

Sol 64K with 2 slot Helios and Panasonic monitor/TV. Software includes Word Wizzard, Mail Master, Fortran(which has been debuged and works!), Regular Basic, Basic org. at D000 and Test Programs on Disk and tape. Original Fortran, Basic and Sol System Manuals , PTDOS manual zeroxed. Condition excellent with hardware updates where indicated. This, as my other two Sols has been entirely trouble free. Reason for selling is simply that I have one Sol at work and two at home. $2700.00

Dick Black,
2721 St. Paul Street, Baltimore Md., 21218, 301-366-1075

FOR SALE-

Centronix model 700s, uppercase unidirectional parallel with 14" tractor. Clean, like new 450.00

Dick Black,
2721 St. Paul Street, Baltimore Md., 21218, 301-366-1075

As-is Sol-20, needs work, (no response when turned on), 2X16 KRA boards. Make offer.
Sanyo 9" video monitor, dual quad density Micropolis drives. Make offer.
Louis T. Jemas 1840 Church St. San Francisco, CA
(415) 772-4375 days, (415) 824-3229 eves.

FOR SALE

PT Sol, Helios II System with 9" Sanyo VM 4092 Monitor and Okidata 110 Printer. Printer is RS 232 with tractor.Has Sol Rev.D with 48K memory, PTDOS 1.5 (mod 2), Serial port driver for printer, Extended Disc basic and several games, etc. Excellent condition and full documentation.

H.W. Chamberlain, 1253 Hawthorne St. Alameda, CA 94501 (415) 521-0221

For Sale

Several Sols available at Comart Limited, Little End Road, Eaton Socon, St Neots, Huntingdon, Cambridgeshire PE19 3JG
Telephone 0480 215005 Telex 32514 Comart G.

FOR SALE

IBM model 2315 disc packs, fits several other drives. $110.00 each. Tape, 1/2 inch wide, on 10 inch reels, in boxes. $6.00 each.
Bob Schaeffer (907) 344-0082 6-11 PST
PO Box 4-1983, Anchorage, AK 99409

FOR SALE -- SOL 20 48K -- two North Star drives, Heathkit monitor. Lots of software including CP/M, CBASIC, ECB 'disc' basic, games, etc. $1200. Works fine. Upgrading to 8" system. Texas Instruments 743 printer/terminal $750. Gerry Fricke, 435 S. Charter, Monticello Il 61856. (217-762-7143).

WANTED

Documentation for a Wang model 701A printer.

Gary E. Lunsford Home (803) 766-0909
1888 Halo Lane Work (803) 554-5565 X 427
Charleston, SC 29407

WANTED

A program for SOL-20 (32K Memory-TapeCassette Storage) RTTY to run on 8080.
Fred Saluna WA6WQZ (415)228-2817 Home-leave message on recorder
(415)645-0088 Office-leave message

WANTED

PTC 3 P+S, working board. Urgent
Mark Berger
1305 Oxford St. Berkeley, CA 94709, (415)843-3214

WANT AD

I am looking for a cassette based version of SOL-FORTH for sale at a reasonable cost. I have purchased the Mitchell E Timin version of Forth, for running on a CP/M 2.2 system.

Mick Kerwick
17Chapel St,
Carrick-on-Suir
Co. Tipperary
IRELAND

Proteus needs a 4 KRA manual for Vol. 6 of the Encyclopedia Processor Technica. If you have one, please send it to us. We will make a copy and return it ASAP.

WANTED

Walnut sides for SOL-20 and power supply.
Bob Schaeffer (907) 344-0082 6-11 PST
PO Box 4-1983, Anchorage, AK 99509

28

TABLE OF CONTENTS

COMPLETE PROTEUS CATALOGUE IN CENTER SECTION

HIGHLIGHTS OF THIS ISSUE

* McVideo 24x80 upgrade really works: Page 1.
* Floppy/winchester/magtape/clock -- all in one slot: Page 1.
* PerSci, the queen of the floppy disks, for $200: Page 5.
* New Proteus catalog: center insert.
* IT'S READY! Encyclopedia Processor Technica: Page 8.
* Make your own 16KRA delay line: Page 13.

And more, and more....

PROTEUS/NEWS

A news journal for owners and users of Processor Technology Corporation computer equipment. Published by Proteus, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483,USA, telephone (415) 368-2300.

Submit items for publication to Proteus News, Attn: Stan Sokolow, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483, USA. Please make submissions as camera-ready as possible by using a fresh, black ribbon and typing single-spaced.

From:
Proteus
1690 Woodside Road, Suite 219
Redwood City, California 94061-3483
USA

James D. McElroy
2826 Crest Ave. North
Allentown, PA

18104

PROTEUS / NEWS

AN INDEPENDENT NEWSLETTER FOR OWNERS AND USERS OF PROCESSOR TECHNOLOGY CORPORATION COMPUTERS

FORMERLY SOLUS NEWS

1982 - SECOND QUARTER PUBLISHED BIMONTHLY BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA

VOL. 5 # 2
SINGLE ISSUE...$7.50 (US)
SINGLE ISSUE...$9.50 (FOREIGN)

Software Tutorial

THE UCSD P-SYSTEM : A UNIVERSAL OPERATING SYSTEM

by Stan Sokolow

--

Synopsis: The UCSD p-System is a programming and operating system that can be adapted to any computer. It is one of the present operating systems vying for its share in the marketplace and gaining acceptance from microcomputer manufacturers. The p-System's portability among dissimilar processors helps the user to maximize the useful life of his software investments. This article describes the p-System (version IV full adaptable system) so the reader may appreciate its features and weaknesses.

--

Wouldn't it be great if your present computer could run any program written for the Apple, the IBM Personal Computer, or any other computer yet to be designed? Then your computer would never really be obsolete, and you would continue to find software being developed for it.

Sol owners should be quite familiar with the transitory world that microcomputing has been. It seems that just about everything in this field was fantasy yesterday, is the latest-and-greatest today, but will be obsolete tomorrow. How can a microcomputer owner help soften the impact of this rapid change in technology?

One way the computing industry has attempted to do this is by use of transportable software. In the beginning, there was only machine language; then symbolic assembly language; then FORTRAN, COBOL, and other higher level languages. But the various dialects of these languages available from different computer manufacturers made it difficult to transport a working program from one commputer system to another dissimilar system.

So, standards were agreed upon for some of the languages. This helped when you took a program to another computing system, but it was not a guarantee of trouble-free portability because there's more to the software than just the language. The operating system and its conventions are just as important as the language itself.

For example, suppose you have a program which creates its own temporary work-files. The name the program assigns to the file is built into the program. But suppose that name is valid on the system where the program was developed, but invalid in some other system because it uses an illegal character? The program will have to be modified, and that may be impossible for the end-user who generally does not have access to the source code.

CON'T ON PAGE 2

WHAT'S NEW?

Datamation magazine July 82 issue has a big article on privacy in home information systems (interactive cable-tv or telephone-based networks). It has some interesting facts and ideas: the FTC came up with 60 possible consumer services for such systems, in 8 categories: home banking, shop-at-home, information services (databases), security services (fire alarm, medical emergency alert, etc), instant opinion polling, home study, special entertainment selections (porno, etc.), and organizational fund raising (targeted to religious, ethnic, etc). The article has a broad overview of the issues and legislation.

Alpha Systems Corporation has announced a "virtual floppy disk" controller for 5 1/4" Winchester disks. This device makes a mini-Wini look like one or two floppy disk drives to your floppy disk controller, but with lots more tracks. Thus you can add Winchester to your existing floppy system with a minimum of hardware and software changes. For example, a 5.5 megabyte drive could appear as two logical floppy drives with nine 1024-byte sectors per track, and 153 tracks per drive. The virtual floppy controller can be jumpered for various Winchesters and to act like an 8" or 5 1/4" floppy. It can be connected to your existing floppy disk "daisy chain" just like more floppies. Alpha Systems Corporation, 711 Chatsworth Place, San Jose, CA 95128, telephone (408) 297-5583.

Destek has announced a "starter kit" for its S-100 network interface board. The Desnet (as the proprietary network hardware is called) can connect up to 350 devices separated by as much as 3000 feet of cable into a 2 megabit-per-second network. A variety of cable types are supported (baseband, braodband, fiber optics, etc). Data can be transferred from node to node faster than from floppy to computer. Network interfaces are available for a variety of computer bus structures, all compatible with the same coaxial network. The two-node starter kit consists of two S-100 boards (one for each node), two coaxial cable taps, cable (RG-59U coax), manual, and software (including source) on CP/M diskette (8" S.D.). Price $1295.

A network operating system with many advanced features for network system architecture will be released soon by Destek.

Destek, 2111 Landings Drive, Mountain View, CA 94043, (415) 968-4593.

VisiGroup is the VisiCalc users group. They also are interested in users of other spread sheet computer systems. Write to VisiGroup, P.O. Box 254, Scardsale, NY 10583.

Steven Wozniak, the co-founder of Apple Computers, has gone back to college (U.C. Berkeley). Some new ideas have apparently diffused into his brain and he has invested 10 megabucks out of his own piggy bank into his new "UNUSON CORPORATION" to put on the "US" festival in southern California. Rock music, country western, homebrew computers,

CON'T ON PAGE 18

CON'T FROM PAGE 1

So, the operating system should be considered just as important for portability as the language (actually more important for the end-user). The machine underlying all of this software is really not important at all, as long as it can do the job.

After all, the hardware is just a vehicle for executing programs and interfacing with the real world outside of the machine. Does it really make any difference what kind of processor is executing the program? Sure, there are differences among computers in their speed, memory capacity,etc., but when you get down to it, they're all about the same. Why not let them all understand precisely the same instructions so that programs written for one will work on all the others?

That's the idea behind the UCSD p-System. It is a portable operating system with several language processors, all written for a computer that doesn't exist. This pseudo-computer (the "p-machine") is what gives the p-System its high degree of portability, as I'll explain in a moment.

This operating system was originally developed by the University of California at San Diego as a tool for teaching computing to their students. But the project grew and was licensed to Softech Microsystems as the exclusive vendor. They have enhanced the system and its language processors, but the universality of the operating system was impeded by the lack of acceptance by computer manufacturers, until fairly recently.

I think the announcement by IBM that the IBM Personal Computer would be available with either the MS-DOS, or CP/M-86, or p-System operating systems as options has given the p-System a real boost. Before this, both Apple and NorthStar offered versions of the p-System, but now that IBM has give its endorsement, it seems that more manufacturers are offering it too. I've noticed that Zenith, Texas Instruments, Phillips, and others have announced its availability.

Well, enough about Softech's marketing; what's this business about the p-System running on a pseudo-machine? Here's how it works. The entire system -- operating system, file manager, language compilers, assemblers, etc. -- is all written in a pseudo-machine language called "p-code". These instructions are stored in an area of memory and are treated as data by a program (elsewhere in memory) that is written in the machine's actual hardware language (known as "native code"). This program interprets each instruction, one by one, and performs the equivalent instructions in the native code. For example, the operation code 162 means "add integers", and it adds the top two integers on the p-machine stack, leaving the result on the top of the stack. Each type of microcomputer has its own specific p-code interpreter, but they all act upon the same p-code instruction set in the same way.

Thus, the entire operating system and all of its related programs can be adapted to run on any machine simply by writing (or buying) a p-code interpreter for that machine. Any computer, from a simple 8-bit microcomputer all the way to the most gargantuan mainframe, can run the same p-code program.

If you accumulate a large (expensive) body of software that runs on your present machine, and you decide you can't live without that jazzy new computer someday, you don't have to start all over paying for equivalent software (assuming it's even available any longer). All you buy is the p-system interpreter for your new machine. The same object (p-code) programs will run on the new computer without change.

Perhaps even more important is the fact that most any object code developed for a newer p-code host machine will run on your present machine.

Of course, you have the problem of transmitting the programs out of the old machine and into the new one, but that's just a matter of making the right hookups, usually through a serial or parallel port. If you're lucky enough that both machines can read the same disk formats, then it's a trivial task.

CP/M's Portability

Now, wait a minute, you may be thinking. Isn't CP/M just the same? After all, you can get CP/M that runs with the Apple (it has a 6502 microprocessor) or with any 8080 or Z-80 microcomputer (like the Sol), or the IBM Personal Computer (an 8088), or the Hewlett-Packard HP-87 (using a proprietary processor), right?

Well, actually when CP/M runs on the Apple computer, what you have is a Z-80 microcomputer board plugged into the Apple. The Z-80 board disables the Apple's 6502 and takes over. A similar thing is done in the other computers, too. So, really CP/M (the 8-bit version known as CP/M-80) only runs on one family of microprocessors (8080/8085/Z-80).

Sure, Digital Reseach has or is coming out with CP/M-like operating systems for other microprocessors (8086, 68000, etc) but OBJECT code written for one of these will not fly on any other microprocessor. Without recompiling the source code, you can't transport your software. And end-users generally don't get the source code.

So, if you purchase software that runs under CP/M-80 in 8080/Z-80 code, and then you get a Z-8000 processor, you must say goodbye to all of your packaged software and buy new packages written in Z-8000 code. And what if your favorite package (word processor, etc.) which you know like the palm of your hand just isn't available for your new computer? You're out of luck.

The p-System Hosts

In contrast, the p-System and any object program written with it will run on any computer, via the p-code interpreter. Softech presently has p-code interpreters available for the following computer processors: 8080/Z80, PDP-11/LSI-11, 8086/8088, 6502, 6809, 68000, 9900. Other vendors support p-System interpreters for different processors (such as Z-8000).

What You Get

In addition to the p-code interpreter and the operating system itself, you get a macro assembler for your host-machine, two editors (screen-oriented and line-oriented), various utilities, and library routines. The language compiler of your choice is available separately. Softech sells compilers that translate the following languages into p-code: UCSD Pascal, FORTRAN, BASIC. There is also a file-conversion utility that lets you convert CP/M files into a form usable by the p-System, and vice versa. Other languages, text processors, editors, etc. are available from independent vendors.

Softech sells the present p-System (version IV) in several formats. Two are of interest to Sol owners. There is a CP/M adaptable version that will bootload using a CP/M command and then take over, replacing CP/M but using the CP/M BIOS (basic I/O system) routines to communicate with the I/O devices.

And there is the full adaptable system, which requires the installer to write a simplified BIOS for communicating with the I/O devices, but which does not require CP/M at all. (The CP/M

CON'T ON PAGE 3

CON'T FROM PAGE 2

adaptable system assumes that you have 128-byte sectors on your disks, so if you want to take advantage of your machine's ability to use more efficient sector sizes, you should use the full adaptable system.)

I am presently working on adapting the full-adaptable system to my Sol with Helios disk, and when it's ready, I'll make it available through Proteus. The Versatile Disk Controller (Proteus Item M5) will be next in line. For other disk controllers, you'll have to adapt it yourself or find someone who's done it already. I'll say more about the adaptation/installation process later.

Internal Architecture of the p-Machine

The programmer or end-user doesn't really need to know much about the internal details of the p-machine, but it is nevertheless documented in the "Internal Architecture Guide" that comes with the system.

When the p-machine is running, there are several important areas of memory (RAM) space. The p-code interpreter itself resides somewhere, usually at the top or bottom of the memory space installed. The I/O routines comprising the BIOS (or SBIOS simplified BIOS) also are resident. The remaining space is available for the p-machine Stack, the Heap, and the Codepool. Here's a typical layout, although location of the interpreter and BIOS can be changed.

```
-----------------------      high address
Operating system kernel
 (always resident)
-----------------------
      Stack
-----------------------
         .
         .
         .
-----------------------
      Codepool
-----------------------
         .
         .
         .
-----------------------
       Heap
-----------------------
       Bios
         &
    Interpreter
-----------------------    low   address
```

The Stack is where operands are placed. The p-codes generally refer to the top-of-stack and top-of-stack-minus-one locations for the operands. Local storage is also allocated automatically here as procedures are entered.

The Heap is a user-programmable dynamic memory allocation area. The programmer can ask for more space on the Heap for storage of variables, arrays, records, etc., and when finished with it, the program can return space back to the Heap.

The Codepool contains segments of the currently running program. The operating system automatically will overlay older segments with segments that are needed, although frequently used segments can be locked-in to prevent excessive overlaying ("thrashing"), thus speeding up execution time.

Programs in the Codepool are dynamically relocated when necessary, to make room for incoming segments. Separately compiled units of programs are automatically linked when they are loaded (that is, the addresses of entry points into the new unit are patched into the calling program). A mixture of p-code and relocatable machine native-code can be in the Codepool. These features allow very large programs to run, much larger than could fit into the machine all at once.

Concurrent Tasks

Moreover, the p-System has built-in features which allow more than one program to be executing at the same time. This is know as multi-programming (multiple programs for one processor), not to be confused with multi-processing (multiple processors executing different or cooperating programs). Each concurrently executing program is known as a "task" or "process".

The operating system automatically allocates separate code and stack space for the active processes, and provides the primitive operations needed to start and stop tasks and coordinate the sharing of resources using "semaphores". (See the book by Per Brinch Hansen: Operating System Principles, Prentice-Hall, 1973.) Hardware events, such as the occurrence of an interrupt, can signal the system so that tasks which were waiting for those events can be activated to respond accordingly. Explanation of the concurrency features of the p-System would take too much space, so you'll have to get the documentation to learn more about it yourself.

Varieties of I/O

The p-System extends the limited input/output capabilities of the standard Pascal language. I/O can be done record-by-record, using the GET and PUT functions, for files that have a defined record type. Untyped files (those without a declared record structure) can be read or written a block at a time. (In the p-System, all disk devices are treated as though they stored data in blocks of 512 bytes each. The BIOS is responsible for blocking and unblocking the data onto the actual device sectors.)

Text files are considered to be streams of ASCII characters, with a few modifications. There is a hidden 2-block header containing formatting information used by the screen-oriented editor. Also, to support the conventional indented style of Pascal, the leading spaces on each line are removed and replaced with a special character and the count of compressed spaces.

The other type of I/O supported is Screen I/O. This is done with a collection of library routines that give the programmer control over video display devices in a hardware independent manner. Using these routines, the program can find out the screen size (height and width in units of character cells), where the cursor is currently located on the screen, whether it can up-scroll, down-scroll, whether it is fast or slow, etc. It can also perform various control functions, such as clear screen, clear line, cursor movement, etc, in a machine-independent and terminal-independent manner. The Screen I/O is easily customized to your terminal or built-in display by a Setup utility.

Getting Specific on I/O

The BIOS (Basic Input/Output System) supports the common I/O devices: console, printer, several disks or other block-structured devices, remote communication port. It also allows the installer to provide his own custom User Device routines for any type of device not anticipated in the standard I/O devices.

CON'T ON PAGE 4

CON'T FROM PAGE 3

All I/O operations set a completion code (IORESULT) that can be tested automatically or by explicit instuctions in the user's program. There are separate codes for such errors as: CRC error, illegal device number, illegal operation on device, device not online, write protect, buffer overflow, illegal block number, and so on.

I/O can even be done in physical-sector mode, rather than logical-block mode, by setting a bit in the parameters passed to the I/O procedures. This allows the high-level languages to do things where assembly language is normally needed.

Asynchronous I/O is supported simply by setting a bit in the I/O calling parameters. (It is not implemented in the stock version of the adaptable system.) This allows the program to continue doinq useful work while an I/O operation is in process. Before the data is used, the completion of the asynchronous I/O must be tested.

Volumes

Each diskette has a volume name stored in its directory. When the user refers to a file, he can prefix the file name with a volume name, like this: "MASTER:ADDRESSES". This example refers to the file "ADDRESSES" on the diskette volume "MASTER". When the system boots up, it scans the disk drives to see which volumes are online. This means that user programs can refer to the diskette "MASTER" no matter which drive it is in, and the system will access the proper one.

When the system accesses a directory, it checks the volume name, and if it is different than expected, it asks the user to insert the desired volume into the drive. Once a drive is found to have an incorrect diskette in it, the system will always check the volume name before using the diskette on that drive, since the drive is considered "questionable" during the execution of the current program. This is a level of user-friendliness and fail-safety that is way beyond what CP/M offers. In business applications, where the inexperienced user may have the wrong diskette inserted, or where diskettes must be swapped in and out for normal processing, it is nice to have the system do this checking.

The Pascal Compiler

The gem of the p-System, and its original raison d'etre, is the Pascal language compiler. Pascal is a language developed to teach contemporary concepts in programming style. As a teaching device it was adequate, but it lacked features that were needed for real-world programming of stand-alone computer systems. The UCSD Pascal project added features to the original language, such as external file names, string data types, segmentation of programs into separately compiled units, and so on. This dialect of Pascal is known as UCSD Pascal.

Pascal has become a favorite language among the academics because of its elegant simplicity and richness of expression. Pascal programs are more readable than those written in BASIC, for example, are more easily maintained, and are less likely to contain logical or careless errors.

You can read more about UCSD Pascal in a variety of Pascal textbooks and reference guides. Along with the p-System and Pascal compiler documentation, Softech sends the following books when you order the system:

The UCSD Pascal Handbook, by Randy Clark and Stephen Koehler, Prentice-Hall, 1982, ($15.95 retail).

Beginner's Guide for the UCSD Pascal System, by Kenneth L. Bowles, Byte Books subsidiary of McGraw-Hill, 1980, ($11.95 retail).

You may also find useful:

Introduction to Pascal (Including UCSD Pascal), by Rodnay Zaks, Sybex, 1981, ($14.95 retail).

Algorithms + Data Structures = Programs, by N. Wirth, Prentice Hall.

Software Tools in Pascal, by Kernighan & Plauger, Addison-Wesley.

Other Languages

To broaden the appeal and utility of the p-System, both BASIC and FORTRAN compilers are available. These provide the standard core of instructions plus extensions to allow the programmer access to p-System features. The operating system allows the programmer to mix segment routines in various languages, so each language can be used where it is easiest.

The Adaptable Assemblers

Since the p-System can be adapted to so many different host machines, it was necessary to make an assembler for each processor type. All of these assemblers, however, are really versions of one universal assembler that takes advantage of the common features in all machine assembly languages. The specific machine operation code mnemonics and syntax error messages are contained in tables that the assembler reads from disk files. Each p-System comes with an assembler and tables for the host machine. Other assemblers for different machines can be purchased. Since all of the assemblers are distributed in p-code, any p-System machine can assemble native-code programs to run on any other p-System host.

The assembler provides advanced features, such as macros, relocatable object code, linkage to Pascal procedures, access to variables in Pascal programs, etc. It is a one-pass assembler, which is unusual but fast.

Native Code Generators

To help speed up critical routines and yet avoid the use of assembly language, there is an optional utility available to convert p-code into the host's actual machine code. Native code takes more space than p-code, but it executes faster because no interpretation is needed. At the moment, not all host processors have native code generators available, but there is one for the 8080, the Z80, and for the 8086/8088.

CP/M Compatibility

The p-System cannot run programs that were intended for the CP/M system, unless they are adapted of course. But a utility program called "XenoFile" will transfer p-System files to a format usable by CP/M, and vice versa. In this way, you can pass data and even source code between the dissimilar systems.

TURTLEGRAPHICS

A collection of hardware-independent graphics routines, called the Turtle Graphics unit, is available as an option. It provides routines for controlling the background, drawing figures, altering old figures, scaling, saving figures on disk files and retrieving them, etc. Multiple view-ports ("windows") on the screen may be defined. Images may be in color or monochrome.

CON'T ON PAGE 5

CON'T FROM PAGE 4

Print Spooler

Spooling is a term that has its origin back in the days of the 2nd generation mainframe computers, which used the "IBM card" for most data and program input. Rather than wasting the time of the expensive mainframe during the reading of cards (at only a few hundred cards per minute), a smaller computer was used to put the card images onto a spool of magnetic tape for later input to the mainframe. Likewise, output from the mainframe was directed to a tape drive for later off-line printing on the cheaper mini-computer. Hence, the idea of storing printer output data on a magnetic storage medium came to be known as spooling.

The p-System can automatically manage a queue of files waiting to be printed, and a Print Spooler utility is available to do this. It is sophisticated enough to allow the user to add files to the queue, interrupt the printing between files to let a higher priority file go directly to the printer, delete files from the queue, suspend or abort printing of files, and so on. All of the printing of queued files goes on concurrently with execution of the user program. If a user program tries to print directly onto the printer, it will be suspended until the printer is available between files.

To support the Spooler, the system must have a keyboard driven by interrupts. That is, pressing a key must generate a hardware interrupt to a service routine that signals the p-code interpreter (by calling a certain location) that a key is ready. The Sol does not normally have a keys-pressed interrupt, but it is not difficult to add one. I plan to write an article on it after I've completed the installation of the p-System on my Sol/Helios.

Console Set-Up

The p-System provides video-oriented console I/O that is the same regardless of terminal type. That is, there are subroutines available in the system for all of the common video control functions (clear screen, home cursor, direct cursor addressing, etc.) and the routines use data given at the time the system is customized by the installer of the operating system.

Since there is no standard for these control functions, each terminal has its own way of taking instructions for these fuctions. (That is, the control characters have different meanings among terminals.) In CP/M, each application program must be customized to the peculiarities of the terminal, whereas in p-System applications, the customization is done when the system is installed, not each time a new program is purchased.

The p-System comes with its own Set-Up utility that interactively obtains data about the terminal from the user. The system installer executes the Set-Up program once, and from then on all programs can interact properly with the terminal. In CP/M, each application must provide a Set-Up routine or menu of terminals.

Clock

The system provides a standard way of reading a hardware clock. The clock is assumed to generate 60 ticks per second, and the clock routine counts ticks in a 32-bit binary accumulator. The count can be obtained by calling a built-in function.

The clock, unfortunately, is not assumed to represent the real time-of-day. Instead, it is merely an elapsed time counter. The p-System uses it in some way to decide when to update diskettes from more recent data still in buffers in RAM.

The clock is NOT used for time-sharing interrupts. Switching among tasks executing concurrently is not done by hardware time-slicing. Rather, this is done by software so that tasks will switch only at a clean break between p-code instructions.

Duplicate Directories and Error Recovery

Diskettes are very good at retaining data, but one in a while the data is damaged. This can happen through careless handling, by a hardware malfunction, or by a faulty program going wild.

If the damaged data happens to be the directory to the disk volume, in one swift blow you can lose access to all of the files, although they may still be in perfect condition on the diskette. The p-System allows you to protect against that somewhat. You can designate that a volume should maintain a duplicate copy of the directory on other tracks. Each time a change is made in the main directory, the secondary directory will also be updated.

If the directory is damaged, you can execute a utility command that copies the secondary to the primary directory, hopefully restoring access to the files.

Other utilities are provided for scanning a diskette and marking bad blocks so they will not be used by the system, and for other kinds of error recovery and file reconstruction if necessary.

User's Group

The UCSD p-System Users' Society (USUS) can be reached through USUS Secretary, P.O. Box 1148, La Jolla, CA 92038, U.S.A. Individual dues are $20 (U.S.) per year. You'll get a journal with letters, articles, and ads from various vendors who produce compatible software for the p-System.

Application Programs

With the growing acceptace of the system, I expect more applications to become available. This will surely be dependent upon the installed user base, as they say. It's only worthwhile to make a p-System version of a program (such as VisiCalc) if there is a large enough body of p-System users.

I've heard that Softech is publishing a list of vendors and application programs.

Ads in the USUS News show that text-processors, ISAM (indexed-sequential access method) packages, screen form generator, spelling corrector, and other applications or tools are available.

Documentation

As I mentioned, the documentation includes two books about Pascal and the p-system. In addition, there are several large softcover manuals on installation and use of the system. A guide to the internal architecture (p-machine operation codes, p-machine pseudo-structure, file formats, directory format, etc.) gives the sophisticated user the inside story.

I'm sure there are omissions in the documentation that will show up once I get into heavy use of the system, but compared with the documentation sent by Digital Research for CP/M, this documentation is a gold mine. It's professional in appearance, too, unlike the CP/M-80 pamphlets.

CON'T ON PAGE 6

CON'T FROM PAGE 5

Price

The prices may be different now, but prior to June 18, 1982, the adaptable p-System cost $375, and the Pascal compiler cost another $375. This includes the operating system, file handler, interpreter, two editors, p-code debugger, assembler for your host machine (both 8080 and Z-80 assemblers are included in the 8080/Z80 package), UCSD Pascal debugger, linker, utilites, and documentation package.

FORTRAN-77 compiler is also $375; BASIC is $225; a package of assemblers for various processors (Z80, 8080, LSI-11/PDP-11, 6502, 6800, 6809, Z8, and 9900) is only $375. A second assemblers package (8086/8088 and 68000) is $125. The 8080/Z80 native code generator is $150; the print spooler is $50; Xenofile (CP/M interchange utility) is $50; TurtleGraphics is $75.

Run-time packages, without the compilers, are available for bundling into turnkey programs.

Recommendation

On paper, the p-System looks great so far. Since my only experience with it is from an early version lacking much of the sophistication of the version IV system, I can't yet speak from personal hands-on experience. When I have some more experience with it, I'll write again.

It does have some good points going for it:

1. It is available for more different microprocessors than any other operating system. Object code for one will run on all, as long as it doesn't contain any native-code routines.

3. In a local network, dissimilar machines can all execute the same object code (p-code).

4. You don't have to throw away more money buying new versions of packaged programs when you move to a new machine.

5. P-code takes less space than native code, so larger programs will fit into memory.

6. The system manages internal memory (RAM) allocation and automatically loads, links, and relocates subroutines as needed on a dynamic basis.

7. It provides a more fail-safe operating environment than CP/M.

There are some drawbacks, too:

1. Execution of p-code is slower than execution of native-code, so the compiler, assembler, etc. don't really run as fast as they could.

2. At the moment, there are less applications available for it than for CP/M, but as more p-systems are sold by computer manufacturers, this will improve.

MODIFIED HYTYPE I PRINTER DRIVER
by Dean Mazur

Processor Technology committed a few blunders with their printer driver which was distributed with their SOL-Hytype interface card. Enclosed is a listing of a very simple adaptation of this original driver. Enough changes have been made that it should not be confused with PT's, but labels and port abbreviations have been retained so that those with access to the PT manual for the interface card may easily compare this driver with the PT version.

1) The original driver did not properly ignore control characters which were not defined in the table of special characters. Code has been added to ignore these.

2) There were several superfluous equates in the beginning of the code, some of which were used by the PTDOS version. An example of these were 'higher level commands'. Hmpf. All superfluous equates have been removed and actual values used instead in arithmetic sections.

3) The entry point assumes the character is in register C, as per CP/M, and not in register B. Also, the formfeed character is properly defined as 0Ch, not 0Bh.

4) The most serious errors were in the error handling sections of the driver. The portion labeled 'this will confuse the hell out of people' in the original listing did a number on the author as well. Because a restore sequence in the Hytype can take several seconds, depending on the position of the carriage when the check condition occurred, the WOS timer usually expires before the restore completes, issuing a 'Controller not responding'. For this reason, I decided to modify the error handling so that either a CHECK or NOT RESPONDING error condition will issue a message directly to SOLOS (note SOUT equate) and set the error flag. Then, on the next call to the driver, a restore sequence is forced, but not top margin, and ALIGN FORMS is sent to screen so operator can clear jam and set up paper at top edge. LOAD key is then depressed to force form feed, which jumps to top margin. Subsequent calls to driver are processed normally. (Note that your BIOS must be modified to trap the LOAD key before the 8th bit is stripped, and to send a formfeed to the LST: device without returning a character - see second listing, which is an extract from one version of our BIOS)

5) Because of the size of the driver, we located it in 2708 EPROM above SOLOS and the Morrow disk controller firmware, but used spare SOLOS RAM for the few status bytes and the several variables needed to store x & y coordinates. The third listing is a very crude PROM burner which should work with most 2708 PROM boards if the equates are changed to match yours. (Note that 1K must be burned at a time, to allow adequate heat dissipation for the 2708 type device.)

6) The PT Hytype controller card is a very simple, reliable unit and appears to be well designed. If your system can tolerate the space needed for the driver, or if you can figure out a way to switch in a bank of 2708 when needed, it will give you total control over the innermost workings of the printer. (except software control of ribbon lift for two color printing)

7) Finally, the user is free to alter margin values, deltas (used to define Elite or Pica pitch) and options. I suggest that you write a short, separate program to configure these interactively so that you don't have to fool around with DDT or SID to patch them. Also, change the equates so that the driver defaults to your most commonly used values if they differ from the ones we use.

Regards,

Dean Allen Mazur

Dean A. Mazur
805 W. 8th St.
Port Angeles,WA 98362

CON'T ON PAGE 7

CON'T FROM PAGE 6

```
CP/M MACRO ASSEM 2.0    #001    2708 EPROM BURNER PROGRAM

                ;*************************************************************
                ;*                                                           *
                ;*                 2708 EPROM BURNER PROGRAM                 *
                ;*                                                           *
                ;*************************************************************
                        TITLE   '2708 EPROM BURNER PROGRAM'
 0100                   ORG     100H
                ;
 0000 =         WBOOT   EQU     0000H   ;WARM BOOT POINT
 0005 =         BDOS    EQU     0005H   ;BDOS ENTRY POINT
 0002 =         CONOUT  EQU     0002H   ;CONSOLE OUTPUT FUNCTION
 0009 =         PSTRING EQU     0009H   ;PRINT STRING FUNCTION
 000A =         READBUF EQU     000AH   ;READ CONSOLE BUFFER FUNCTION
 E000 =         PROM    EQU     0E000H  ;BASE OF PROM IN USER SYSTEM
 000D =         CR      EQU     0DH     ;CARRIAGE RETURN
 000A =         LF      EQU     0AH     ;LINE FEED
                ;
                ;       LOAD SP AND PUT OUT SIGN-ON MESSAGE
                ;
 0100 31C002            LXI     SP,STACK
 0103 11FA01            LXI     D,SIGNON
 0106 CDEF01            CALL    PRINT   ;PRINT MESSAGE AFTER CRLF
                ;
                ;       GET SOURCE ADDRESS AND PERFORM HEX CONVERSION
                ;
 0109 111902    SOURCE: LXI     D,PROMPT1       ;'ENTER SOURCE ADDR' PROMPT
 010C CDEF01            CALL    PRINT
 010F 0E0A              MVI     C,READBUF       ;READ BUFFER FUNCTION
 0111 119402            LXI     D,CONBUF        ;POINT TO THE BUFFER
 0114 CD0500            CALL    BDOS
 0117 3A9502            LDA     CONSIZE         ;GET INPUT LENGTH
 011A FE00              CPI     0
 011C CCDD01            CZ      BADINP          ;COMPLAIN IF NO INPUT
 011F CA0901            JZ      SOURCE
 0122 FE05              CPI     5
 0124 CCDD01            CZ      BADINP          ;COMPLAIN IF 5 CHARS INPUT
 0127 CA0901            JZ      SOURCE
 012A 219602            LXI     H,CONLIN        ;POINT TO LINE PORTION
 012D 010000            LXI     B,0             ;CLEAR BC
 0130 4F                MOV     C,A             ;PUT SIZE IN LOW ORDER BYTE
 0131 09                DAD     B               ;SLIDE POINTER TO END + 1
 0132 3624              MVI     M,'$'           ;MARK IT TO KEEP CP/M HAPPY
 0134 119602            LXI     D,CONLIN        ;POINT TO LINE PORTION AGAIN
 0137 210000            LXI     H,0             ;CLEAR HL
 013A 1A        HCONV:  LDAX    D               ;GET CHAR FROM LINE PORTION
 013B FE24              CPI     '$'             ;CHECK FOR END OF LINE
 013D CA5301            JZ      TARGET
 0140 29                DAD     H
 0141 29                DAD     H
 0142 29                DAD     H
 0143 29                DAD     H
 0144 CDCC01            CALL    HEX
 0147 D4DD01            CNC     BADINP
 014A D20901            JNC     SOURCE          ;TRY AGAIN
 014D 85                ADD     L
 014E 6F                MOV     L,A             ;MOVE IT IN
 014F 13                INX     D               ;BUMP POINTER
 0150 C33A01            JMP     HCONV
                ;
                ;       GET TARGET PROM NUMBER AND CONVERT IT
                ;
 0153 229B02    TARGET: SHLD    SADDR           ;SAVE SOURCE ADDRESS
 0156 113602    TARGET1:LXI     D,PROMPT2       ;'ENTER TARGET' PROMPT
 0159 CDEF01            CALL    PRINT
 015C 0E0A              MVI     C,READBUF
 015E 119402            LXI     D,CONBUF
 0161 CD0500            CALL    BDOS
 0164 3A9502            LDA     CONSIZE         ;GET INPUT SIZE
```

```
0167 FE01               CPI     1
0169 C4DD01             CNZ     BADINP
016C C25601             JNZ     TARGET1
016F 119602             LXI     D,CONLIN        ;POINT TO LINE PORTION
0172 1A                 LDAX    D               ;GET CHAR
0173 FE30               CPI     '0'             ;WAS IT A 0 ?
0175 DCDD01             CC      BADINP
0178 DA5601             JC      TARGET1
017B FE38               CPI     '8'             ;WAS IT AN 8 ?
017D D4DD01             CNC     BADINP
0180 D25601             JNC     TARGET1
0183 D630               SUI     30H             ;BETWEEN 0 & 7 SO REMOVE BIAS
0185 47                 MOV     B,A             ;SAVE PROM NO. IN B
               ;
               ;        CONVERT PROM NUMBER TO A TARGET ADDRESS
               ;
0186 110004             LXI     D,1024          ;SIZE OF A 2708
0189 2100E0             LXI     H,PROM          ;POINT TO BASE OF PROM
018C 97                 SUB     A               ;0 INTO A
018D B8        LOOP:    CMP     B               ;COMPARE TO PROM NO.
018E CA9601             JZ      BURN            ;DONE WITH CONVERSION
0191 3C                 INR     A               ;BUMP COUNT
0192 19                 DAD     D               ;ADD 1024 TO BASE ADDRESS
0193 C38D01             JMP     LOOP
               ;
               ;        NOW WE CAN BURN THE PROM
               ;
0196 229E02    BURN:    SHLD    TADDR           ;SAVE THE TARGET ADDRESS
0199 110004             LXI     D,1024          ;SIZE OF A PROM
019C 19                 DAD     D
019D 7C                 MOV     A,H             ;GET HIGH ORDER BYTE
019E 329D02             STA     TESTVAL         ;SAVE IT FOR LATER
01A1 010000             LXI     B,0             ;PASS COUNT = 0
01A4 2A9E02    BLOOP:   LHLD    TADDR           ;GET TARGET ADDRESS
01A7 EB                 XCHG                    ;NOW IN DE
01A8 2A9B02             LHLD    SADDR           ;SOURCE ADDRESS
01AB 7A        PLOOP:   MOV     A,D             ;GET COUNTER VALUE
01AC E5                 PUSH    H               ;SAVE IT FOR LATER
01AD 219D02             LXI     H,TESTVAL       ;POINT TO TEST VALUE
01B0 BE                 CMP     M               ;COMPARE WITH TEST VALUE
01B1 E1                 POP     H               ;RECOVER TARGET POINTER
01B2 CABC01             JZ      NEXT            ;NEXT PASS IF IT MATCHES
01B5 7E                 MOV     A,M             ;GET DATA BYTE
01B6 12                 STAX    D               ;ZAP THE PROM
01B7 23                 INX     H               ;BUMP SOURCE POINTER
01B8 13                 INX     D               ;BUMP TARGET POINTER
01B9 C3AB01             JMP     PLOOP           ;CONTINUE
01BC 03        NEXT:    INX     B               ;BUMP PASS COUNT
01BD 78                 MOV     A,B
01BE FE04               CPI     4               ;1K PASSES YET?
01C0 C2A401             JNZ     BLOOP           ;KEEP GOING IF NOT
               ;
               ;        TERMINATION MESSAGE
               ;
01C3 117102             LXI     D,TERMMSG       ;SAY BYE-BYE
01C6 CDEF01             CALL    PRINT
01C9 C30000             JMP     WBOOT           ;BACK TO CP/M
               ;
               ;        UTILITY SUBROUTINES
               ;
01CC D630      HEX:     SUI     48              ;REMOVE ASCII BIAS
01CE FE0A               CPI     10
01D0 D8                 RC                      ;IF LESS THAN 9
01D1 D607               SUI     7               ;IT'S A LETTER
01D3 FE10               CPI     10H
01D5 C9                 RET                     ;WITH CARRY IF F OR LESS
01D6 0E02      PUTCHR:  MVI     C,CONOUT        ;CONSOLE OUTPUT FUNCTION
01D8 5F                 MOV     E,A             ;CHAR TO SEND IN RIGHT PLACE
01D9 CD0500             CALL    BDOS
01DC C9                 RET
01DD 115302    BADINP:  LXI     D,ERRMSG        ;POINT TO ERROR MESSAGE
```

CON'T ON PAGE 8

CON'T FROM PAGE 7

```
01E0 CDEF01             CALL    PRINT
01E3 C9                 RET
01E4 3E0D       CRLF:   MVI     A,CR            ;CARRIAGE RETURN
01E6 CDD601             CALL    PUTCHR
01E9 3E0A               MVI     A,LF            ;LINE FEED
01E8 CDD601             CALL    PUTCHR
01EE C9                 RET
01EF D5         PRINT:  PUSH    D
01F0 CDE401             CALL    CRLF
01F3 D1                 POP     D
01F4 0E09               MVI     C,PSTRING       ;PRINT STRING FUNCTION
01F6 CD0500             CALL    BDOS
01F9 C9                 RET
                ;
                ;       STRING DATA AREA FOR CONSOLE MESSAGES
                ;
01FA 3237303820SIGNON: DB       '2708 PROM BURNER - Version 1.0$'
0219 456E746572PROMPT1:DB       'Enter source address (hex): $'
0236 456E746572PROMPT2:DB       'Enter target PROM  (0 - 7): $'
0253 4552524F52ERRMSG: DB       'ERROR: Incorrect input format$'
0271 4275726E20TERMMSG:DB       'Burn completed...returning to CP/M$'
                ;
                ;       VARIABLE AREA
                ;
0294 05         CONBUF: DB      5       ;LENGTH OF BUFFER
0295            CONSIZE:DS      1       ;SIZE AFTER SUCCESSFUL READ
0296            CONLIN: DS      5       ;BUFFER PLUS 1 MORE FOR $
0298            SADDR:  DS      2       ;SOURCE ADDRESS
029D            TESTVAL:DS      1       ;TEST VALUE (EQUALS 0 FOR FC00 SOURCE)
029E            TADDR:  DS      2       ;TARGET ADDRESS IN 2708
02A0                    DS      32      ;16 LEVEL STACK
02C0            STACK:  END
```

ED. NOTE: Modified Hytype 1 Printer Driver program added to EPT Volume 11.

---

by Allen Fincher

DOCUMENTATION FOR "FORM" FILE

The file "FORM" on this disk is a BASIC program for printing "Port Connection Designations" forms. The form allows you to document the connections between your computer and a peripheral. It may be filled out as follows:

Put an "x" in PARALLEL or SERIAL boxes.

If serial, fill in the data rate (i.e. 300 BAUD) and the data configuration (i.e. 8 data bits, no parity, 1 stop bit).

In "Type Connectors", fill in the port connector type (i.e. DB-25) and then the peripheral connector type (i.e. edge connector, DB-25, etc.).

Next, fill in the addresses of the data and status ports, in hexadecimal.

In the Peripheral Connection column, write in the peripheral pin connector pin number that is connected to the corresponding Port Connection pin listed. Then, in the second column, show the direction of the signal (grounds would go both ways).

Next, describe the function of the wire.

Lastly, if applicable, write in the hex mask necessary to test the bit represented by each wire. For example, if Port Connection 8 was connected to Peripheral Connection 3 as a busy flag (direction: from peripheral to port) and the flag could be tested as bit 6 of a status port read, then the Hex Mask would be 40H on Port Connection row 8. You also may wish to indicate if the bit to be tested is active low. Only connections that can be read by reading the status port need have a hex mask value given, unless you have a control device as the peripheral.

The form will take up the entire length of an 11 inch long page. If the paper is fan-fold, then more than one copy of the form can be printed, the perforations will be skipped.

```
 10 REM:  FILENAME:  FORM
 20 REM
 30 PRINT "\K"
 40 PRINT TAB(10);"'PORT CONNECTION DESIGNATIONS' FORMS PRINTER"
 50 PRINT
 60 PRINT TAB(23);"Version 2.2  1/7/81": PRINT : PRINT
 70 INPUT "How many forms do you want?  ",F
 80 REM
 90 REM:  SET OUTPUT FILE TO PRINTER DRIVER
100 REM
110 SET OF="PR80"
120 SET LL=80
130 REM
140 REM:  PRINT THE FORMS
150 REM
160 FOR N=1 TO F
170 GOSUB 280
180 NEXT N
190 REM
200 REM:  RESET OUTPUT FILE TO VDM DRIVER AND LINE LENGTH TO 64
210 REM
220 SET OF=#0
230 SET LL=64
240 END
250 REM
260 REM:  FORM PRINTING ROUTINE
270 REM
280 PRINT TAB(28);"PORT CONNECTION DESIGNATIONS"
290 PRINT
300 PRINT TAB(17);"PERIPHERAL:";: LET L=30: GOSUB 620
310 PRINT : PRINT
320 PRINT TAB(17);"PARALLEL []      SERIAL []"
330 PRINT TAB(35);"DATA RATE:";: LET L=6: GOSUB 620
340 PRINT TAB(51);"BAUD"
350 PRINT TAB(35);"DATA CONFIGURATION:";: LET L=25: GOSUB 620
360 PRINT : PRINT
370 PRINT TAB(10);"PORT ADDRESSES:";
380 PRINT TAB(40);"TYPE CONNECTORS:";: LET L=10: GOSUB 620
390 PRINT "/";: LET L=12: GOSUB 620
400 PRINT : PRINT
410 PRINT TAB(12);"DATA:";: LET L=5: GOSUB 620
420 PRINT "H";
430 PRINT TAB(24);"STATUS:";: LET L=5: GOSUB 620
440 PRINT "H"
450 GOSUB 690
460 PRINT TAB(7);"PORT    I ---> I PERIPHERAL I";
470 PRINT TAB(50);"FUNCTION";TAB(71);"I  HEX"
480 PRINT TAB(4);"CONNECTION I <--- I CONNECTION I";
490 PRINT TAB(71);"I  MASK"
500 GOSUB 690
510 REM
520 REM:  PRINT FORM LINE
```

CON'T ON PAGE 13

BULLETIN SYSTEM FOR PROTEUS VIA RBBS

by Alastair Preston

Dear Stan,

Can you or other members tell me if there is a Bulletin Board System regularly used by Proteus members? One reason I bring up this question is that such a system could be used for rapid submission of messages, letters and articles for PROTEUS/NEWS. Our local computer club makes extensive use of it's own RBBS and two others currently in operation in this city, to receive such material for the club's monthly newsletter. The editor can then transmit all the messages etç., to the local university's computer system for final editing, collating and printing out of masters for the printer who does the production copies. I don't know how you handle that end of the newsletter production, but I think that the ability to send you material via an RBBS or similar set-up would be quite helpful, at least for those of us who have modems.

4 10335 123 St.,
Edmonton, Alberta,
Canada, T5N 1N5
14 July 1982

Yours truly,

AlPreston.

Alastair Preston.

(Ed. note to Alastair Preston:
The bulletin board systems are great, but they don't seem appopriate for what we've been doing in Proteus. Our subject matter is not urgent. Besides, they don't allow transmission of drawings, so that makes it hard to send schematics, etc. I don't use any of them myself, but I'm sure some members do. Perhaps they can give us some opinions and recommendations. Thanks for the suggestion.)

SUPER-PHANTOM FOR SOL-20

by Fr. Thomas McGahee

Dear Stan,

I hope to get some more stuff off to you soon, especially some stuff on how to make a memory decode circuit to deactivate RAM that may be addressed where you have disk controller boards and the like. Unfortunately, not all disk controllers provided an on-board generated PHANTOM to deactivate RAM. I also hope to send along a fix for the NORTHSTAR controller board that will make it generate the proper PHANTOM signal. I have all this stuff up and running... th problem is getting the time to write it up neatly and get it submitted to PROTEUS. I don't want to send in something that is half-baked.

Stan, I would really appreciate it if you could keep this article together as a whole. I know this isn't always possible, but it sure makes it easier on the reader when it is all in one place.

Keep up the good work. Say hello to your secretary for me. She was very pleasant and most helpful when I called while you were out. She sounds like a really nice person.

Fr. Thomas McGahee
Don Bosco Tech.
202 Union Ave.
Paterson, NJ 07502
May 17, 1982

Sincerely yours,

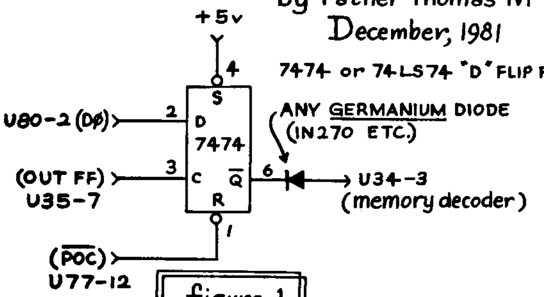
Fr. Thomas McGahee

Fr. Thomas McGahee

LET YOUR SOL — 20 HAVE A FULL 64K OF RAM AND SOLOS AT C000

SUPER-PHANTOM FOR SOL-20

by Father Thomas McGahee
December, 1981



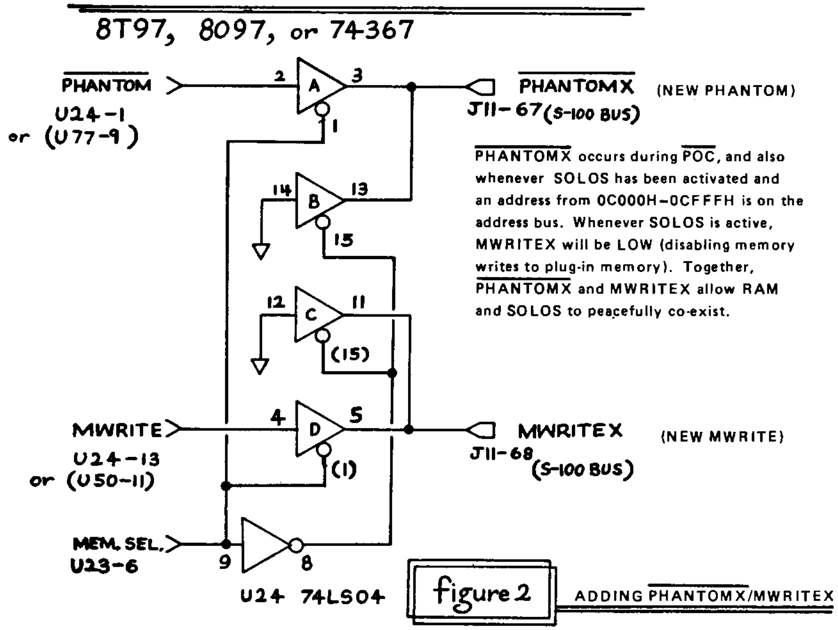
TO ALLOW RAM TO OVERLAY
0C000H - 0CFFFH,

```
MVI A,01
OUT 0FFH
```

TO RE-ACTIVATE SOLOS,

```
MVI A,00
OUT 0FFH
```

figure 1 ADDING THE (SOLX) PORT ((OFFH))



$\overline{\text{PHANTOMX}}$ occurs during $\overline{\text{POC}}$, and also whenever SOLOS has been activated and an address from 0C000H-0CFFFH is on the address bus. Whenever SOLOS is active, MWRITEX will be LOW (disabling memory writes to plug-in memory). Together, $\overline{\text{PHANTOMX}}$ and MWRITEX allow RAM and SOLOS to peacefully co-exist.

figure 2 ADDING $\overline{\text{PHANTOMX}}$/MWRITEX

CON'T ON PAGE 10

CON'T FROM PAGE 9

May 17, 1982

Dear SOL Brothers,

In the May/June/July/August 1981 issue of PROTEUS (the SOL user's newsletter), Stan Sokolow mentioned a method devised by Jim Spann and Chuck Athey for making the SOL vanish. Unfortunately it isn't as simple as they thought. Fortunately it isn't much harder either! Herewith I present a method that overcomes the problems associated with the Spann/Athey method. This modification is currently running on my SOL, so I know it works. It allows a SOL computer to have a full 64K of contiguous RAM, and yet still have access to the regular SOLOS ROM/RAM and VIDEO RAM. It is totally compatible with all previous hardware/software, since on power-up it comes up with the SOLOS ROM/RAM/VIDEO activated as usual. Under software command, however, plug-in ROM/RAM can be made to overlay the SOLOS area from CØØØH-CFFFH.

The Spann/Athey modification involves tricking the memory decoder into ignoring addresses from CØØØH-CFFFH by pulling the [G1] enable line of U34 LOW upon receipt of an OUT FF instruction. The method they proposed almost works. It DOES deactivate the memory decoder. Unfortunately, this is not enough to provide us with true bank-switching. First of all, PHANTOM only occurs during the "four-phase-wonder" following Power On Clear. It does NOT occur when memory from CØØØH-CFFFH is addressed. Secondly, even if PHANTOM DID occur at this time, it would not prevent RAM occupying the same address range from being WRITTEN INTO when SOLOS is active. Most memory boards only use the PHANTOM line to inhibit READING. WRITES are still allowed. While this may not be a problem for the ROM portion of SOLOS, it IS a problem for the SYSTEM RAM area and the VIDEO RAM area!!!

What my modification does is:

1) Deactivate on-board memory (using Spann/Athey method).
2) Create PHANTOM whenever SOLOS is active.
3) Inhibit off-board MWRITE whenever SOLOS is active.

THEORY OF OPERATION AND WIRING

[Use schematic X-16 of SOL MANUAL]
As shown in my Figure 1, A 74LS74 (or 7474) is connected as a one-bit port. The SET pin is disabled by pulling it to +5, and the RESET pin is connected to POC (Power On Clear). The DATA pin is connected to the DØ line, and the CLOCK pin is connected to the FF output of the Output Port Decoder. There is a very good reason for choosing the FF port instead of the FA port as suggested by Stan. If the FA port were used, then sending ØØ or Ø1 out to that port could inadvertantly do such bad things as turn off the tape recorder.

Programs like Richard Greenlaw's TAPEDISK would not like this at all! Using port FF sidesteps all the possible hidden problems that might otherwise arise. I tend to be the cautious type. I used port FC in my original design, but have since changed to port FF to avoid any possible conflicts with Bob Hogg's 8Øx24 modification board for SOLs. (The potential problem here is that he uses port FC for the BELL function. You don't really want the overlay to be inadvertantly activated or de-activated by sending out a BELL code, now do you?) I have one of Hogg's 8Øx24 McVIDEO boards running on my system. It is beautiful, and by using the FF port, I can have my BELL and SUPER-PHANTOM, too.

There is one other difference between my Figure 1 and the Spann/Athey circuit. I added in the GERMANIUM diode because otherwise whenever pins 3 or 11 of U22 were LOW and pin 6 of our 74LS74 was HIGH, a fight would ensue. Using the diode eliminates this problem. By the way, piggy-backing the 74LS74 onto U49 is the way I got power to pins 7 and 14. When piggy-backing, bend all the leads of the 74LS74 out, except pins 7 and 14. Tin leads 7 and 14. Remove U49, and while it is UPSIDE-DOWN, tin just the SHOULDERS of pins 7 and 14. DO NOT GET SOLDER ONTO THE PART OF THE PIN THAT GOES INTO THE SOCKET! Turn the 74LS74 upside down on the table (make sure you know which is pin 7). Turn U49 upside down and rest it on the 74LS74 such that their 7 and 14 pins match up. Spot solder pins 7 and 14 together. Then I did the signal wiring. ALL WIRING SHOULD BE DONE USING WIRE-WRAP WIRE. I attach wire-wrap wire in the following manner. I wire-wrap right onto the pins of the 74LS74, and then solder the connection. I then cut the wire to the proper length. Once all wires are attached to the 74LS74, I strip about 1/8 inch of insulation off of the other ends, bend the exposed wire portion over in half, and tin. Once all wires are tinned, I plug U49 back into its socket, and solder all wires to their proper ICs. I remove the IC in question, turn it upside-down, and tin the SHOULDER of the proper pin. I then spot solder the proper wire in place. Keeping the IC upside-down helps keep the solder only on the shoulder.

[There is an empty socket at location U1Ø3. You can add the 7474 there if you prefer. It makes for a cleaner layout. Since you will be putting a 14 pin socket in, don't forget to insure that BOTH power pins are connected. One of them will have to be jumpered in, since the space was originally designed for a 16 pin socket. Make sure Ground is at pin 7, and +5 at pin 14.]

[Use pages X-15 and X-16 of SOL MANUAL]
Refer to my Figure 2. Whenever MEM SEL is HIGH, indicating that SOLOS is ACTIVE, the B and C tri-state buffers gate LOWS onto the PHANTOMX and MWRITEX bus lines. This informs OFF-BOARD memory that READING and WRITING is not allowed. Whenever MEM SEL is LOW, indicating that SOLOS is inactive, then the normal PHANTOM and MWRITE signals are put on the bus via tri-state buffers A and D. Because of the PHANTOMX and MWRITEX signals, off-board RAM/ROM can coexist with SOLOS RAM/ROM. Which one will respond to the addresses from CØØØH-CFFFH is determined by the one-bit port that we installed using the 74LS74.

After a Power On Clear, the SOLOS RAM/ROM will respond. The off-board RAM/ROM can be made to overlay SOLOS by sending out the following instructions:

```
        MVI     A,Ø1
        OUT     ØFFH
```

SOLOS can be made to regain control by using the following:

```
        MVI     A,ØØ
        OUT     ØFFH
```

I used a 74367 as the tri-state buffer, and soldered it at the empty position marked [U82] on the SOL PC Board. (An alternate position is U1Ø3, or piggy-back it onto an 8T97). In any case, pin 8 is ground, and pin 16 is +5 volts. Please take careful note of the following: ALL ON-BOARD PHANTOM AND MWRITE SIGNALS ARE LEFT INTACT. ONLY THE BUS SIGNALS BECOME PHANTOMX AND MWRITEX. If you have an empty position, then solder in a socket to hold the 74367. Initially solder just pins 8 and 16. As wire-wrap wire is connected to a pin, carefully solder it in place. Using a pair of small needle-nose pliers, I carefully wrap the wire with just one or two turns before soldering. Space is limited, so be neat. Connect pins 12 and 14 of our 74367 to ground (pin 8).

Next, isolate the on-board PHANTOM line from the J11 bus. Using page X-3, locate pin 67 of J11. Notice that it connects to a feed-through on the left side of J11. CUT THE RUN GOING TO THIS FEEDTHROUGH ON THE TOP OF THE BOARD. On the bottom of the board, solder a wire from this feed-through to pin 2 of our 74367. Daisy-chain from pins 13 and 3 of our 74367 to pin 67 of J11 (bottom of board). Next to pin 67 of J11 is pin 68. On the bottom of the board, cut the run that goes to this pin, and connect pin 68 to pins 5 and 11 of our 74367. Using page X-3, locate pin 68 of J11. Notice the run that connects to it coming from the left side of J11 on the TOP of the board. CUT THIS RUN. This isolates MWRITE from the bus, but also from some on-board stuff that needs it, so locate the feedthrough next to R2Ø that connects to this run that we just cut, and solder a wire into it. (Put wire on bottom of board if 74367 is at U82 position).

CON'T ON PAGE 11

CON'T FROM PAGE 10

Connect this wire to the feedthrough from pin 13 of U24, and ALSO connect a wire from either of these feedthroughs to pin 4 of our 74367. Daisy-chain a wire from pin 6 of U23 to pin 9 of U24 and pin 1 of our 74367. Connect a wire from pin 8 of U24 to pin 15 of our 74367. (If pins 8 & 9 of U24 are not free, then you'll have to piggy-back a 74LSØ4 to get an inverter. Some of you may have already made use of this "free" inverter).

Re-assemble your SOL and verify that everything works normally.

USING THE NEW FEATURE

In a CP/M environment, you would want to change the BOOT program to deactivate SOLOS so RAM can be used instead. The simplest way is to place a MVI A,Ø1 / OUT ØFFH at the very beginning of the BOOT program ON DISK. This requires 4 bytes. Some boot loaders may not have enough room. Remember, we are really only concerned about getting DØ high, so if the BOOT routine loads any register with data that has DØ high, then use a register exchange operation to get it copied into "A", and then output it to port ØFFH. That will use 3 bytes. The important thing is that this must be done BEFORE the BOOT routine starts to load the DOS into memory.

The BIOS must also be patched to allow the I/O routines to activate/deactivate SOLOS. The most sane approach is to write a SOLON and SOLOFF routine like these:

```
SOLX     EQU     ØFFH     ;SOLX = NAME OF PORT.
ONX      EQU     ØØ       ;ØØ = TURN SOLOS ON.
OFFX     EQU     Ø1       ;Ø1 = TURN SOLOS OFF.

SOLON:   PUSH    PSW      ;SAVE "A".
         MVI     A,ONX    ;ØØ MEANS TURN ON SOLOS.
         OUT     SOLX     ;DO IT.
         POP     PSW      ;RECOVER "A".
         RET

SOLOFF:  PUSH    PSW      ;SAVE "A".
         MVI     A,OFFX   ;Ø1 MEANS TURN OFF SOLOS.
         OUT     SOLX     ;DO IT.
         POP     PSW      ;RECOVER "A".
         RET
```

Now all references to things like "CALL SINP" should be changed to "CALL XSINP", and these "X" routines take on the following general form:

```
XSINP:   CALL    SOLON    ;ACTIVATE SOLOS.
         CALL    SINP     ;CALL SOLOS SINP.
         CALL    SOLOFF   ;DEACTIVATE SOLOS.
         RET
```

By using this approach, you should be able to make all the changes in the BIOS with a minimum of hassle, and with the maximum of understanding. The overhead is really quite small, as most BIOS programs will only access SOLOS for about five or six routines such as SINP, SOUT, SDROT, SSTAT, etc..

POINTS TO CONSIDER

In a CPM system, the BIOS may NOT be in the range CØØØH-CFFFH. The rest of the DOS may reside there, but not the BIOS. The reason for this restriction is quite simple: if the switching on and off of SOLOS takes place within the BIOS, and the BIOS turns itself OFF, then control is lost! One other restriction is that in CPM the STACK may not reside from CØØØH-CFFFH, since the stack is used for CALLs and RETurns. Should you find this to be a problem in your system, it can be easily cured by patching SOLON and SOLOFF to include a simple stack swap routine. SOLON would switch to a temporary stack (it need only be about 24 bytes deep), and SOLOFF would recover the original stack. Thus, the temporary stack (placed in "safe" RAM), would be used to stack CALLs initiated within SOLOS.

Here are examples of routines that swap STACKS:

```
SOLX     EQU     ØFFH     ;SOLX = NAME OF PORT.
ONX      EQU     ØØ       ;ØØ = TURN SOLOS ON.
OFFX     EQU     Ø1       ;Ø1 = TURN SOLOS OFF.

SOLON:   PUSH    PSW                ;SAVE "A".
         MVI     A,ONX              ;ØØ MEANS TURN ON SOLOS.
         OUT     SOLX               ;DO IT.
         POP     PSW                ;RECOVER "A".
         SHLD    TEMPH              ;SAVE HL.
         POP     H                  ;... GET RETURN ADDRESS
         SHLD    RETX               ;... AND STORE IT.
         LXI     H,ØØ               ;ZERO OUT HL.
         DAD     SP                 ;HL = STACK-POINTER.
         SHLD    OSTACK             ;SAVE OLD STACK ADDRESS.
         LXI     SP,TSTACK          ;SET UP TEMPORARY STACK.
         LHLD    RETX               ;RECOVER RETURN ADDRESS.
         PUSH    H                  ;PUT IT ON STACK.
         LHLD    TEMPH              ;RECOVER ORIGINAL HL.
         RET

SOLOFF:  PUSH    PSW                ;SAVE "A".
         MVI     A,OFFX             ;Ø1 MEANS TURN OFF SOLOS.
         OUT     SOLX               ;DO IT.
         POP     PSW                ;RECOVER "A".
         SHLD    TEMPH              ;SAVE HL.
         POP     H                  ;... GET RETURN ADDRESS
         SHLD    RETX               ;... AND STORE IT.
         LHLD    OSTACK             ;RECOVER OLD STACK ADDRESS.
         SPHL                       ;MAKE IT CURRENT.
         LHLD    RETX               ;RECOVER RETURN ADDRESS.
         PUSH    H                  ;PUT IT ON STACK.
         LHLD    TEMPH              ;RECOVER ORIGINAL HL.
         RET

TEMPH    DW      ØØ                 ;HL STORAGE.
RETX     DW      ØØ                 ;RETURN ADDRESS.
OSTACK   DW      ØØ                 ;OLD STACK ADDRESS.
LSTACK   DS      24                 ;LENGTH OF STACK IS 24 BYTES
TSTACK   DB      ØØ                 ;[TSTACK NEED NOT BE IN BIOS].
```

USING ELECTRIC PENCIL II

Some programs make direct use of the video memory. ELECTRIC PENCIL II is one such program. Since there is no way of letting this program switch the video memory on and off as it needs it, we have a problem. One solution is to restrict the ELECTRIC PENCIL II to memory below CØØØH. The ELECTRIC PENCIL II uses the address at locations 6 and 7 to determine how much memory it can use as a buffer. The way we fake out the ELECTRIC PENCIL II program is to change the address at these locations so that it refers to location BFFD. Then at BFFD we place a JMP to the actual base of the BDOS (whose address USED to be at 6 & 7). This can easily be done within the BIOS, but is also easily done by patching the ELECTRIC PENCIL II file. I will explain how to patch the disk file.

Using DDT, load the ELECTRIC PENCIL II file. In my system, the file loads from 1ØØH to 18ØØH, which is a total of 23 records. (1*16 + 8 - 1). My file starts off with a bunch of NOPs and then a JMP 147H. I changed the first three NOPs to a JMP 18ØØH. Then at 18ØØH I patched in the following:

CON'T ON PAGE 12

CON'T FROM PAGE 11

```
PATCH:  LHLD    6               ;GET BDOS ADDRESS INTO HL.
        MVI     A,ØCFH          ;IF BDOS IS BELOW DØØØH...
        CMP     H               ;...THEN SKIP THE PATCH.
        JNC     147H
        MVI     A,ØC3H          ;CODE FOR A JMP.
        STA     ØBFFDH          ;STUFF IT.
        SHLD    ØBFFEH          ;NOW STUFF THE BDOS ADDRESS.
        LXI     H,ØBFFDH        ;POINT TO OUR VECTOR.
        SHLD    6               ;MAKE IT CURRENT.
        JMP     147H            ;END OF PATCH.
```

I then typed a CONTROL/C to get back to the system, and typed "SAVE 24 PEN.COM". The reason for using 24 rather than 23 is that the patch also had to be saved. This is a "smart" patch, and it will automatically adjust the BDOS vector ONLY IF NEEDED. In other words, the ELECTRIC PENCIL II file will then work properly for both old and new systems. Neat, huh?

Please note that for use with the ELECTRIC PENCIL II, the BIOS should leave SOLOS ON at all times, so your existing BIOS is used without changes. Just make up a CPM system that has the system size set so that the CCP loads in at DØØØH or above. BY THE WAY, THE ABOVE TECHNIQUE IS ALSO USEFUL EVEN IF YOU HAVE NOT MADE THE HARDWARE CHANGES, SINCE IT EFFECTIVELY "SKIPS OVER" THE SOLOS ROM/RAM AREA. You MUST have sufficient RAM from DØØØ-UP to handle the CCP/DOS/BIOS of CPM, however. As you can see, because the CPM operating system is in RAM above the SOLOS area, this leaves a full 48K of RAM available for the ELECTRIC PENCIL II program. For most of us this is still a significant increase, since the disk operating system now resides ABOVE SOLOS instead of below it.

Similiar techniques can be used with other programs that use direct video access. For example, many SOL users running WORDSTAR use it in the memory-mapped mode.

This article was originally submitted January 4, 1982. When it did not appear in the last issue of PROTEUS, I re-submitted it with a few minor changes to make the instructions clearer. So far I have modified about ten SOLs, and all have worked perfectly. In addition, a number of members of the Amateur Computer Group of New Jersey have made the modifications and verified that the instructions are complete. As I mentioned before, the modifications here are compatible with the new McVIDEO board from Bob Hogg. Incidentally, I opted to use his Dual Personality module the following way: Regular SOLOS at CØØØ (this is the original 64x16 SOLOS), and Bob Hogg's 24x8Ø enhanced SOLOS monitor ALSO AT CØØØ. He makes many versions of the ROM available... all at the same cost. But you must be explicit in telling him what versions you want! In my system I put the 24x8Ø enhanced SOLOS into the socket labelled "FØØØ". Since it is actually at CØØØ, I had to make a minor change to his instructions that came with the 8Øx24 display. I simply did not connect either the yellow or green wires that he specifies to connect to pins 5 & 2 of U22. Instead, I carefully taped them up so they wouldn't touch anything.

The reason why I chose not to use a system relocated to FØØØ is simple: Switching takes place within the BIOS, and if the BIOS overlays the ROM or systems ram or video ram, then it would lose control and crash when it attempted a switch. Besides, on one of my systems, my disk controller resides at F8ØØ, making an FØØØ system impossible. With a CØØØ system, the only thing you have to make sure of is that the portion of your BIOS that performs the SOLOS I/O are above CFFF. If you have a disk controller sitting at an inconvenient location, you can always go with an FØØØ system. In this case, you can get some use out of the overlay RAM by loading utilities at 1ØØH and then relocating them from FØØØ up. Just make sure such utilities don't try to directly access the monitor routines or they will crash. By this time you get the basic idea. Carefully look over your system needs before deciding on a CØØØ or FØØØ system.

By the way, if your disk controller can generate a PHANTOM signal whenever it is selected, GREAT! Many controller boards only generate PHANTOM on a boot. Check your schematics. If your board CAN or DOES generate a PHANTOM signal whenever it is selected, then you will have no problems with a 64K memory board. If your controller DOES NOT generate such a PHANTOM signal, then you will have to deselect blocks of memory, or modify your controller board to provide the proper signal, or build a simple memory decoder circuit that will generate a PHANTOM signal whenever the proper addressing range is detected. I have done all of the above at one time or another. I hope to write a short article for PROTEUS soon outlining how I did it with a NORTHSTAR controller, and how to build an address range decoder circuit that will work with ANY controller. (It consists of one 16 pin IC, and mounts ENTIRELY on the bottom of the mother board). Most likely you will find it in this issue of PROTEUS.

I hope that you find the hardware/software techniques discussed here to be of value in allowing your SOL system to become even more useful. Remember: when your SOL powers up, it will come on exactly as usual. ALL your present software will run exactly as it is. BUT, NOW you have the option of using a FULL 64K of RAM and still have all the goodies inside SOLOS available whenever you really want them. Because of the compatability feature, I strongly recommend that SOL users make these modifications. Once again, the SOL has proven itself to be a superior machine. My thanks to Stan Sokolow, Jim Spann and Chuck Athey for coming up with and publishing the original concept. And let's not forget Processor Tech, which had the foresight and good sense to furnish every SOL owner with a complete set of schematics and circuit descriptions. Without these, many of us would never have had the courage to even consider modifying our beloved SOLs at all.

CONTRIBUTED BY
FR. THOMAS MCGAHEE
DON BOSCO TECH.
2Ø2 UNION AVE.
PATERSON, NJ Ø75Ø2
(2Ø1) 595-88ØØ

Incidentally, the school is a tax-exempt charitable institution, and donations of computer-related books and material are greatly appreciated.

Editor's note: Jim Spann described a correct implementation of his mod in (S-100) Microsystems magazine. We've reprinted this in EPT Volume 2, chapter 4 page 61.

LETTERS TO THE EDITOR

...on the McFloppy, McSolos, and Electric Pencil

24MAY82
PO BOX 440357
AURORA, CO 80044
(303) 750-1838

STAN,

AS USUAL, PROTEUS CAME THE OTHER DAY, AND I DIDN'T PUT IT DOWN UNTIL I WAS THROUGH READING IT FROM COVER TO COVER. I WISH IT WERE THICKER. I HAVE A FEW COMMENTS, SUGGESTIONS, AND ADDITIONS THAT I WOULD LIKE TO ADD THIS QUARTER.

HAVING LIVED WITH THE NORTH STAR SINGLE DENSITY(SINGLE DISK) SYSTEM FOR FOUR(4) YEARS NOW, I THOUGHT IT WAS TIME TO UPGRADE TO DOUBLE DENSITY. THE SYSTEM I DECIDED TO GO WITH WAS MICRO COMPLEX'S NORTH STAR LOOK-ALIKE, "MCFLOPPY". THE REASON WAS TWO-FOLD.

(1) FIRST, I WANTED TO BE ABLE TO RETAIN MY NORTH STAR COMPATIBILITY WITH EXISTING SOFTWARE.

(2) SECONDLY, BECAUSE THE EXTRA FEATURES THAT CAME WITH THE PHASE LOCK II CONTROLLER BOARD, AND DOS.

CON'T ON PAGE 13

CON'T FROM PAGE 12

FOR THOSE MEMBERS WHO MAY NOT BE FAMILIAR WITH THIS PRODUCT FROM MICRO COMPLEX, LET ME ELABORATE. THE PHASE LOCK II DUAL DENSITY CONTROLLER IS A NORTH STAR LOOK - ALIKE CONTROLLER BOARD. IT IS PROVIDED WITH IT'S OWN DOS CALLED "MCDOS". THE DOS IS ALSO COMPATIBLE WITH EXISTING NORTH STAR SOFTWARE, HOWEVER ITS INTERNAL STRUCTURE IS CONSIDERBLY DIFFERENT. SOME OF THE FEATURES THAT MAKE THIS BOARD AND DOS AN EXCELLENT CHOICE ARE:

(1) WITH AN OPTIONAL BOOT PROM AND DIP SWITCH THE BOARD IS RELOCATABLE.

(2) IT WILL WRITE USING THE NEW TANDON DOUBLE SIDED DRIVES IN BOTH THE 48TPI AND 96TPI. THIS ALLOWS UP TO 819K BYTES PER 96TPI DRIVE - INCREDIBLE!

(3) THE DOS DOES NOT USE ANY MEMORY ABOVE 2A00, SO ALL NORTH STAR SOFTWARE WILL WORK - EVEN PROGRAMS THAT START AT 2D00.

(4) THE I/O AREA IN MCDOS COMES CONFIGUIRED FOR A "STANDARD" SOL AT C000 (LOAD & GO).

AFTER PURCHASING THE CONTROLLER I DECIDED THAT A SECOND DRIVE WAS NECESSARY ESPECIALLY IF I WAS GOING TO GET CP/M UP, SO I BOUGHT A TANDON TM100-2. THIS IS A 48TPI DRIVE WITH DUAL HEADS SO IT WRITES ON BOTH SIDES OF THE DISKETTE. WITH DOUBLE DENSITY FORMATTING THIS ALLOWS YOU TO PUT ABOUT 409K BYTES ON A SINGLE 5 1/4" DISKETTE - THAT'S ALOT OF STORAGE! THE BOARD AND DISK HAVE BEEN OPERATING FLAWLESSLY FOR OVER A MONTH - I AM VERY PLEASED WITH MY NEW SYSTEMS SPEED, POWER, AND PERFORMANCE.

ANOTHER PRODUCT THAT MICRO COMPLEX MARKETS FOR THE SOL IS THE DUAL PERSONALITY MODULE. I PURCHASED ONE ALONG WITH THE CONTROLLER, AND HAVE FOUND IT TO BE A VERY USEFUL ADDITION TO MY SYSTEM HARDWARE. NOT ONLY DOES IT RELOCATE SOLOS UP TO F000 FOR USE WITH CP/M, BUT IT ALSO HAS AN ENHANCED VERSION OF SOLOS (MCSOLOS) WHICH INCORPORATES MANY NEW COMMANDS. NO MORE LOADING THE NORTH STAR MONITOR TO DO SYSTEM MAINTENANCE AND DEVELOPMENT - IT'S ALWAYS ON-LINE. A LIST OF COMMANDS FOLLOWS THAT ARE AVAILABLE WITH THE NEW "MCSOLOS".

DA - DISPLAY ASCII MEMORY DUMP
DH - DISPLAY HEX MEMORY DUMP
EA - ENTER ASCII CHARACTER STRING
EH - ENTER HEX BYTE STRING
FM - FILL MEMORY
MM - MOVE MEMORY BLOCK
CM - COMPARE MEMORY BLOCK
AR - ADDRESS REFERANCE LOCATOR
FH - FIND HEX WORD OCCURANCES
FB - FIND HEX BYTE OCCURANCES
FA - FIND ASCII CHARACTER PAIR
FC - FIND ASCII CHARACTER SINGLE
TM - TEST MEMORY

ALSO PROVIDED ARE SOME JUMP VECTORS THAT CAN BE MODIFIED WHEN ORDERING FOR A MINIMAL CHARGE. THEY ARE -

OS - VECTOR TO DISK OPERATING SYSTEM @ 2028
WB - WARM BOOT VECTOR TO 0000 FOR CP/M
WP - WORD PROCESSOR VECTOR @ 2A0C
BA - BASIC RE-ENTRY VECTOR @ 2A04
BC - BASIC RE-ENTRY W/CLEAR @ 2A00
BV - BASIC RE-ENTRY NO RESETS @ 2A14

ANOTHER NICE FEATURE IS THAT THE "LOAD" KEY ON THE SOL WILL NOW BOOT NORTH STAR AT E800. THAT VECTOR ADDRESS CAN ALSO BE MODIFIED WHEN ORDERING FOR A MINIMAL CHARGE. THIS MAKES IT A SNAP TO BOOT MY SYSTEM UP IN MCDOS.

ON A DIFFERENT SUBJECT - I NOTICED IN THE LAST ISSUE THAT THERE SEEMS TO BE A LOT OF US OUT THERE THAT ARE STILL USING SOME VERSION OF ELECTRIC PENCIL. PERHAPS THE REASON IS THAT THE ELECTRIC PENCIL IS STILL A FINE SOFTWARE PACKAGE THAT RUNS ON THE SOL, AND MEETS MOST OF OUR WORD PROCESSING NEEDS, OR IF YOUR LIKE ME AND HAVEN'T CONVERTED TO CP/M SO THAT YOU CAN RUN OUT AND PLUNK DOWN $500 TO MICROPRO FOR "WORDSTAR". SINCE THERE SEEM TO BE SO MANY MODIFICATIONS TO PENCIL - WHY NOT TRY AND OBTAIN THE SOURCE CODE FROM MICHAEL SHRAYER. IF THOSE OF US THAT WOULD LIKE TO KEEP E.P. ALIVE WOULD EACH KICK IN SAY $50.00 WE(PROTEUS) JUST MIGHT BE ABLE TO PURCHASE IT. ANY THOUGHTS??? THE REASON I MENTIONED THIS WAS BECAUSE AFTER I HAD MY NEW PHASE LOCK II CONTROLLER RUNNING I TRIED TO LOAD MY DISK VERSION OF PENCIL AND YOU GUESSED IT, IT DIDN'T RUN. THE SINGLE DENSITY VERSION OF E.P. DID NOT USE DOS (SHRAYER WROTE HIS OWN DOS). IT ALSO LOADS AT 0000H.
SO NOW I HAVE BOTH MY OLD SINGLE DENSITY CONTROLLER, AND THE PHASE LOCK II CONTROLLER IN MY SYSTEM ADDRESSED AT DIFFERENT LOCATIONS - JUST SO I CAN RUN PENCIL. ANY IDEAS??????

I DID IMPLEMENT THE PENCIL MODIFICATION BY LEON WINTER, AND ENJOY THE INVERSE VIDEO ALOT - THANKS LEON.

WELL, MY CP/M HASN'T ARRIVED YET SO I'LL GET THIS LETTER OUT THE DOOR. I WILL WRITE ABOUT MY CP/M EXPERIENCES AS SOON AS I GET IT UP AND RUNNING.

THANKS AGAIN STAN, FOR THE EFFORT YOU PUT FORTH ON OUR NEWSLETTER.

BEST REGARDS,
Rick
RICK DOWNS

CON'T FROM PAGE 8

```
530 REM
540 FOR C=1 TO 25
550 IF C<10 THEN  LET T=8 ELSE  LET T=7
560 PRINT TAB(T);C;
570 PRINT TAB(15);"I";TAB(22);"I";TAB(35);"I";TAB(71);"I "
580 GOSUB 690
590 NEXT C
600 PRINT
610 RETURN
620 REM
630 REM:  PRINT BLANK LINE, LENGTH = L
640 REM
650 FOR C=1 TO L
660 PRINT CHR(95);
670 NEXT C
680 RETURN
690 REM
700 REM:  PRINT FORM RULING
710 REM
720 PRINT TAB(4);"-----------+------+------------+";
730 PRINT TAB(23);"----------------------------------+-------"
740 RETURN
```

PascalSET Tools from First Systems Corporation

by Stan Sokolow

I recently responded to an ad for Pascal programming aids called PascalSET. Here's what I learned. The set consists of an execution tracer to aid debugging, a preprocessor to add ADA language features to Pascal, and a Macro preprocessor. The whole package costs $450, or each aid can be bought separately. The software is available for large DEC computers, but is in the process of being converted to run on micros.

First Systems Corporation, 865 Manhattan Beach Blvd., Manhattan Beach, CA 90266. (213) 546-5581.

...on ExpandoRam, ECBASIC to CP/M Conversion, Micro Complex Video Board by John Barber

Dear Stan,

In reference to R Ellingsworth's letter, (V 4, #5/6), I had no problem with an unmodified Expandoram I until I got a Vista disk system. There was a short in U-63 on BUSS 75, and the memory board worked fine until that IC had to be replaced in order to get the disk system running. I have had no problems since making the mods suggested in PROTEUS/News after that happened. It works with Vista.

If extreme low price is not the object, as it was in my case, I would recommend a more compatable format, i.e. North*, or Diskus. It is difficult to get software in VISTA format. TAPEDISK somewhat alleviates this objection.

Here also is a review of the TAD Enterprises Extended Cassette Basic CP/M conversion.

The CP/M conversion for ECB from TAD Enterprises, P. O. Box 257, Hazelcrest, Il 60429, is a true conversion. It comes with a cassette file to which ECB is appended, which modifies I/O to disk instead of cassette. It also comes with a utility that changes input and/or output back to cassette, and introduces several new functions. They allow the change of the CP/M file type, normally *.ECB, a trace function which follows the line numbers, a single disk drive command, a warm boot command to allow the changing of diskettes, and a modification to allow the editing and saving of a submit file.

Saving to disk in T(ext) format permits a file to be read and edited by ED.COM or any other text editor. All in all a very useful feature.

Now for the bugs. I have not been able to save files to cassette in T(ext) format. Since there are two very easy ways to program around this it is no serious defect. One can load ECB from tape, load the file saved in semi-compiled format, and save in T(ext) format, or add Byte by Lew Mosley to PIP.COM. The installation manual says that one can also use the conversion for Basic/5. Since in three or four attempts I was unable to write files after this conversion, I have discarded any intent to use this. It would have only been a bonus anyway. I would use a conversion to 6/2 Microsoft BASIC more, anyway. I never did use BASIC/5 much, anyway.

Over all, the conversion is a fine way to keep software written in ECB alive, and gives one a CP/M based SOL compatable Interpreter at a reasonable price, $49.95. I suspect there may be some problems after conversion to F000, and to 80/24 video format. I think we will have to talk to the author about that.

Speaking of that problem about the Micro Complex 80/60 video mod, most software that uses CP/M uses the BIOS, but I would love to see the mods required to re-INSTALL Wordstar, and I think other files such as T/MAKER will also need to be customised for the new display. If anyone has done the work, please send the results to PROTEUS/NEWS and save everyone repeating the job.

See the latest issue of S-100 MICROSYSTEMS (Jan/Feb 1982) for an article on making SOL vanish. The trick seems to be to put SOLOS and system ram on the internal buss, rather than the S-100 buss and then bank select. With a little more work, it should also be possible to phantom out a memory mapped disk controller, such as North*. I think I would still like the 80/24 screen size, and the Micro-Complex PM seems easier to use.

CONTINUED....

I agree with Lew Mosley. One of the major advantages of a SOL is the ability to exchange software between a large variety of very different systems using the cassette interface. I would not want the Micro-Complex Dual Personality Module without the cassette routines that could be switched to at C000H.

Yours truly

John Barber

(Ed note to John Barber: You can order the McDPM with standard Solos in the C000 version, but most people will just put the cassette routines onto their disk system and load them when needed.)

...Method Wanted for Cataloging Helios Disks

23 Feb. 1982

Dear Stan,

Has anyone come up with some method of catalogueing Helios Disks, somewhat in the same manner as can be done for CPM, using a fine system offered by Richard Greenlaw? One should be able to catalogue, and create a master catalogue together with disk name or number.

The information and letters about the 80 x 24 conversion in Vol.4 # 5 & 6 of Proteus is good news. But there may be some drawbacks, namely, does it preclude the use of any disk drive that has a ROM in the F000 block? Moreover, can one use PTDOS?

As many of our members, I too believe PTDOS is a better operating system than anything else around,in fact makes CPM look positively cumbersome. Too bad that those who have control of the source code do not make use of it to make PTDOS machine independent, so that programs could be written using it, or transposed to it. Of course, we all know that the dictates of the market place have something to do with all this, i.e. is there any money in it? Given the proper marketing that too could be overcome. Witness the fine software that the Basic Computer Groups in Vancouver wrote for WORD WIZARD or MAIL MASTER. A word processor such as WORD WIZARD is hard to beat,and I understand that GENERAL LEDGER has software control to get around PTDOS and make use of 64K.

I am most interested in the outcome of the Helios/Morrow Disk Multiplexer prototype which Ace Computers has, and would like to hear about the outcome of any demonstration you might have seen.

As usual,there are always fine articles and informative letters in PROTEUS, for that we all thank you, the staff and our many faithful contributors and members.

Sincerely,

Frank J. Sanders
28 Alanbrooke Ct.,
Baltimore,Md. 21204

(Ed. note to Frank Sanders:
Good news. A PTDOS version of Greenlaw's cataloging system is in the Helios library for future release.
The 24x80 video conversion can go completely out of the Sol's address space under software control. You can use unused flip-flops to disable your disk ROM when accessing the screen or Solos. Also, you can alter the ROM address decoder on the controller and make it respond to E000 instead of F000.)

...Booting Up the Sol with Northstar

PROTEUS
1690 *WOODSIDE ROAD, SUITE* 219
REDWOOD CITY, CA. 94061

DEAR SIRS;

INCLOSED IS A SELF ADDRESSED - STAMPED ENVLOPE FOR A RESPONSE IF ONE IS POSIBLE. I HAVE BEEN A SUBSCRIBER FOR 2 YEARS NOW AND SEE CONSTANT REFFERENCES TO A SERIES OF BOOKS ON THE SOL PROCESSOR. I HAVE A SOL WITH 32K AND A NORTH STAR DISK SYSTEM. ALL THE EQUIPMENT WAS BOUGHT USED. DUE TO THIS THE INFORMATION I HAVE IS A LITTLE SKETCHY. WHAT I AM LOOKING FOR IS INFORMATION ON USEING THE NORTHSTAR DISK SYSTEM WITH THE SOL-20. I HAVE THE OLDER SINGLE SIDE SINGLE DENSITY UNIT, AND ACCORDING TO WHAT I HAVE READ IT SEEMS A GOOD MATCH WITH THE SOL. THE DRIVE AND BOARD ARE IN VERY GOOD SHAPE, BOTH HAVEING BEEN AQUIRED FROM A DEALER WHO WENT OUT OF BUSINESS. THERE SEEMS TO BE SOME PROBLEM GETTING THE SYSTEM TO BOOT UP, AND I WAS WONDERING IF THERE WERE ANY 'TRICKS' THAT HAD TO BE DONE TO THE SOL OR THE NORTHSTAR BEFORE THE UNITS WOULD WORK TOGETHER. ANY INFORMATION YOUR GROUP HAS ON THIS COMBINATION IS NEEDED. IF THERE IS A CHARGE FOR THIS CONTACT ME OR BILL ME AT THE PLACES GIVEN BELOW.

THANKS FOR ANY HELP. SENCE PTC WHENT DOWN THE TUBES, YOU PEOPLE HAVE BEEN THE ONLY PLACE TO TURN TO FOR HELP. MOST THINGS HAVE BEEN COVERED BY YOUR NEWSLETTER VERY WELL.

YOURS,
Douglas Stowell
DOUGLAS W. STOWELL
1414 *WHIPPORWILL*
GARLAND, TEXAS 75040 214 840 8534

- OR WORK -
STSC, INC. AIRLINES SERVICES
1525 *ELM ST., SUITE* 2660
DALLAS, TEXAS 75201 214 749 1983

...Join the Nevada Colbol Users Group

Nevada Cobol Users Group

5536 Colbert Trail • Norcross, Georgia 30092 • (404) 449-8948

Hello:

I feel that you and other members of your group may be interested in our offering. I have enclosed a membership application for your use. If more are needed please feel free to copy the one enclosed or request more from me directly.

The Nevada COBOL Users Group has been formed to distribute information on applications and routines written in COBOL and to coordinate efforts amoung users developing extentions to the language. Information on routines or programs written for other COBOL languages will be accepted providing they are compatible with Nevada COBOL. Distribution of information will be through periodic newsletters. As the membership expands and the number of contributed articles and programs increase frequency of publication will become more regular with the goal of monthly publication. There will be no charge for the publication until actual production costs are established. Thereafter subscription fees will be based on projected costs which are expected to be minimal. Distribution of the first newsletter is currently set for June 1, 1982.

To discern the type of information you most desire I have enclosed an application for membership which requests information on your background and how you make use of computers or your interest in them. All information that you provide will be held in confidence and will only be used to determine newsletter direction.

Sincerely yours,

Bob

Bob Blum
Chairman

Editor's note: Nevada Cobol was written by one of our members, Chuck Ellis, 600 41st. Ave. #1, San Francisco, CA 94121.

For application to Nevada Cobol Users Group send to above address

..."Sword"; A Mini-Wordprocessor for the Sol

Dennis C.Fait
PO Box 22
Slippery Rock,Pa. 16057
412-794-5243

May 6, 1982

Gentlemen,

What began as a project intended to provide myself with a needed piece of software grew quickly into something I now feel I should offer for sale to Sol/Micropolis users.

That software is a mini-wordprocessor for the Sol. Two versions are available: one disk-based using Microplis' Mod II disk I/O (no other routines in MDOS are used) and one tape-based using SOLOS' tape routines. Both are offered for $50 each.

The wordprocessor, which I call SWORD, has the following features: 1) Text entered as quickly as one can type, with screen wrap-around; 2) Full cursor control using arrow keys; 3) Screen scrolling, up and down, both one line at a time and 15 lines at a time; 4) Insert mode; 5) Deletions using DEL key; 6) Block delete; 7) Sort routine; 8) Clear text buffer with one key; 9) "Home" command displays beginning of text buffer; 10) Named files may be saved and loaded from tape/disk; 11) Files may be appended (disk version only); 12) "Query" command displays status of printer options; 13) Printer linelength, left margin indentation, and line spacing are controlled from the keyboard; 14) Printer commands may be imbedded within the text; 15) Form feed; 16) Page numbering can be toggled on or off.

Although the printer driver was programmed for the Epson MX-80, it will most likely work with other printers. Most characters in the text buffer are sent to the printer port unchanged. (The exceptions are every other escape character and control-V, which are used to configure the printer.)

At this point I would like to modify SWORD for use on other systems using MDOS, but I need more infortion concerning those systems.

By the way, this letter was written using SWORD, including the letterhead (to compose the letterhead, however, another program I've written in BASIC is necessary. That program creates the graphics, which are then transferred to a SWORD file).

Sincerely yours,

Dennis C. Fait

Dennis C.Fait

... Adding More Features to an Upgraded Sol

Dear Stan:

I have completed the installation of Bob Hogg's modification of my SOL to the 24x80 display and am rather satisfied with the conversion. As others have indicated it was not difficult to install and after a couple small problems I was up and running.

I now am interested in adding the following features to my system:

1. Adding another 8K ram. I have an Extensys RAM using the 2108 (H) chip. Does anyone have a lead on 8 chips? I have two RAM boards which each hold 64K but have only enough chips for 48K.
2. I'd like to add another printer using my parallel port. If anyone has added an Epson or other draft printer using the parallel port I'd like to know how they added the software, especially for 1.4 CP/M.
3. I presently use Electric Pencil, version DS-II, and would like to modify it for the 24x80 display. Several articles have been written regarding patching EP but I'd need some more help. Bob Hogg said he'd do it for about $50.00. If anyone else is interested contact me and maybe we can get it done together.
4. I might like to add a new word processing package altogether. Since the SOL is memory mapped I'd like to use that capability. Any comments from other SOL users?
5. I would also like to add a financial planning package. From reading the magazines Supercalc by Sorcim seems interesting. Again, any comments from other SOL users?
6. Next to buying a new system my wish list includes a 5 meg. hard disk. Has anyone had experience with that animal?

Stan, I would appreciate any articles in Proteus regarding screen display programming techniques. For instance, in a mailing list program I would like to set up a format on screen and fill in the information rather than have a menu driven program in which the screen scrolls. I suspect that there must be some technique using the memory mapped video to do this. Being a novice at programming means I need rather clear, step by step instructions.

I would like to mention that I am using my system with a CP/M patch that I worked out using information from Fr. Thomas McGahee to provide a back spacing DEL function and a Diablo driver at 1200 baud. Some information was published previously in Proteus regarding that patch. Several other Proteus members sent me information too. I have been up to my nostrils in other things and haven't been able to reply to their letters. I appreciated the comments but since I was working at getting the bugs out of the patch that was along similar lines that Fr. MaGahee sent me I didn't use the information sent by them. My system uses Micropolis disks and if anyone needs help on that please contact me and I will help them.

I expect to be in California in July and will be shopping for information on the above. If I get a chance I'll stop by for a chat. Hope to see you then.

Chuck Hansing

Chuck Hansing
CH/ep

Chuck Hansing
4741 Hibiscus Ave.
Edina, MN 55435

... A Super Printer for the Bucks

Dear Stan:

At last I can send a decent letter that may be worthy of PROTEUS. I am responding to the call for comments on any new equipment used by PROTEUS users for evaluation. The text you are reading was printed out on the new "Smith-Corona TP-1" daisy wheel printer. It sells for under $900.00 (I paid $799 + cable for mine due to an add typo!) The cheapest I have yet seen it for is from the company I bought mine from, "Micro-printer marketing" (800-523-9859) for $845.

The good news about this new printer, aside from its great looks, is obviously the price. For the poor among us, $845 is a far cry from $2500+ for some of tne more well-known brands of daisy-wheel printers. It is available in either 12 or 10 pitch (pica or elite) and has film ribbons for crisp letters. In addition, it can be purchased either as a serial or parallel port-driven printer. It feeds the single sheets of paper well (the tractor-feed is not yet available for fan-fold paper, I'll let you know about that when it comes out) and is just over-all a super printer for the bucks.

Now- the bad part! Whereas this guy costs one-third the price of the Diablos or their kin, it also takes 3 times as long to print. Approx. 10 c.p.s. For those of you used to a dot matrix that zips along at 30-50 cps, it will seem very slow. For me, I have Sol set at 110 baud, and although the TP-1 is adjustable, found that it couldn't handle anything faster.

That is the only real problem I have found with it so far, and I have dumped out several hours (straight) of term-papers on this guy. It doesn't seem to overheat on long runs, and just keeps plugging at it.

So, if someone is looking for letter quality, and can't afford an expensive printer,(and doesn't mind the printer taking about the same amount of time as a fast human typist), this is a way to go. Ann at Micro-printer Marketing is keeping me posted when other type-styles are availabe, so far they have 10 or 12. This letter was typed in three of the different styles available.
(by the way, Stan, I can't find in my records when my subscription to PROTEUS expires, could you let me know? I want to keep it current)

Paul W. Kittle

PO Box 1285
Loma Linda, Ca. 92354

...HELP WANTED FOR REPAIRING SOLS by Hon Computing Systems

HON COMPUTING SYSTEMS INC.
360 Civic Dr.,
Pleasant Hill, CA 94523
(415) 676-2383

June 30, 1982

Dear Sirs:

HELP! We are a computer dealer who uses the Processor Technology SOL computer as a base for our systems. We are in need of any information that would lead us to someone who is presently repairing SOL units.

We have sold approximately 100 systems in the San Francisco Bay Area; however, information of a technician any where in the United States will be of a great help.

Please send any information to the address above or call us. Thank you for your cooperation and speedy reply.

Sincerely yours,

Galon D. Miller
Service Manager

... A Recommendation for Stetson Electronics

John A. Whiting
230 S. Coronado St. #21
Los Angeles, California
8205.19 1200 hrs. 90057

Greetings, and Salutations:

I just got the latest issue of PROTEUS NEWS and the first thing I noticed (after getting over having had a letter published) was how well my printer reproduced in the issue. My head is now a hat size or two larger.

Anyway, I'm writing this time to announce that I did find a source of the SAA 1027 chip, and the ESM 227 chip and probably a whole bunch of other goodies as well. The source is Bob Stetson, owner of Stetson Electronics, P.O. Box 3008, Nashua, New Hampshire; 03061, (603) 880-4975. The hours are 6:00 p.m. to 9:00 p.m. Boston time.

In addition to mail order parts, he also offers disk drives, software (hopefully, some for SOL, although I didn't see any), and design work for both individuals and companies. Mr. Stetson is extremely helpful and when I discovered that I'd ordered the wrong part, he sent the right one and waited cheefully while I sorounged around for the money. His prices are also reasonable. BASF told me that the part would cost $10.95 (excluding shipping) if they could even get ahold of one; Bob's price was $10.95, including the shipping.

Now all I have to do is get the *@#%&!! drive working. I heartily recommend Stetson Electronics. He doesn't promise more than he can deliver, but he's managed to deliver twice for me, now.

As for PROTEUS NEWS, it was a great issue. I won't mind it becoming a quarterly at all if the subsequent issues are as good as this one. Keep up the good work, folks!

Regards,

John A. Whiting
JAW/sol

Post Script: I enclose Bob's flyer in the hopes that it proves reproducible, and useful.

STETSON ELECTRONICS
P.'O. BOX 3008
NASHUA, N. H. 03051
(603) 880-4975

STETSON ELECTRONICS HAS A WIDE VARIETY OF ELECTRONIC COMPONENTS AND ASSEMBLIES. IF IT'S NOT ON THE LIST, THEN WRITE AND I WILL TRY TO ACQUIRE ONE FOR YOU AT A REDUCED COST.

ITEM	PRICE	SHIPPING
5 1/4'' DISK DRIVE ENCLOSURE WITH SUPPLY	$65.00 EACH	$5.00
5 1/4'' DISK DRIVE WITH ENCLOSURE AND POWER SUPPLY	$175.00	$10.00
5 1/4'' DISK DRIVE NO ENCLOSURE / NO SUPPLY	$125.00	$5.00
5 1/4'' DISK DRIVE SIGNAL CABLES (34 PIN) W / RIBBON CONNECTERS FITS ANY 5 1/4'' DISK DRIVE (SEE BELOW)		
1 DRIVE CABLE	$10.00	$1.00
2 DRIVE CABLE	$15.00	$1.00
3 DRIVE CABLE	$20.00	$1.00
4 DRIVE CABLE	$25.00	$1.00

TAPE OR DISK SOFTWARE AVAILABLE FOR TRS-80 (SPECIFY WHICH)
TRS-80 IS TRADE MARK OF TANDY CORP.
PLEASE SPECIFY LEVEL I OR LEVEL II / MOD I OR MOD III

BOWLING	$15.00	$1.00
CRAPS	$15.00	$1.00
XOLARIAN EMPIRE ...	$15.00	$1.00
GAME PACK DISK ...	$15.00	$1.00

TAPE OR DISK SOFTWARE AVAILABLE FOR APPLE
APPLE IS TRADE MARK OF APPLE COMPUTER CORP.
PLEASE SPECIFY TAPE OR DISK
PLEASE SPECIFY APPLE II OR APPLE II PLUS

XOLARIAN EMPIRE ...	$15.00	$1.00
GRID WARP	$20.00	$1.00

CALL STETSON ELECTRONICS AT (603) 880-4975
WE ARE OPEN FROM 6:00 P.M. TO 9:00 P.M. NIGHTLY (BOSTON TIME)
BUT, BEFORE 9:00PM BOSTON TIME

WE CAN ANSWER QUESTIONS ON OUR GAMES OR COMPONENT PRICING
WE GUARANTEE YOUR SATISFACTION OR YOU RECEIVE A REFUND.

ALL ITEMS AND PRICING ARE SUBJECT TO AVAILABILITY.

...REPLY TO LEONARD KALISH VISTA CBIOS KEYBOARD INPUT PROBLEM

After reading Leonard's fix for the Vista CBIOS in Vol. 4, Issues #5/6, I was reminded about a similar problem I had with my S-100 system. I also had problems with double, and even triple, characters being displayed whenever I typed anything. It turned out to that the strobe pulse that was generated by my keyboard was too wide. That is, the strobe pulse was staying high (my keyboard has an active high strobe, Sol's is active low) so long that it was still high when the program came around to check for another character; so, the program read the data port again and got the same character.

I sounds like the Sol may have the same problem that I had. By looking in the Intel 8080 manual, I found that a CMA instruction takes 4 t-states. Since each t-state is equal to the period of the system clock (1/frequency), the time required for the CMA instruction is 2.00 microseconds for a 2 MHz clock (1 t-state = 0.5 microsecond for a 2 MHz clock). This extra time may be enough delay to allow the Sol keyboard strobe pulse to go high again before the routine in the CBIOS can get to the CINP routine to get another character.

Someone with the necessary knowledge of the Sol's circuitry may be able to determine if a timing capacitor or resistor in the keyboard strobe generating circuit can be changed to a value that will shorten the pulse width enough to allow the CMA instruction to be dropped from programs.

Of course, there is another easy solution. Have all program I/O handled by the routines in SOLOS via the vectors at the beginning of SOLOS. The CINP fix shown in the article could be written as follows:

```
CINP:  CALL  SINP  ;Check keyboard
       JZ    CINP  ;There's nothing there
       ANI   7FH   ;Strip bit 7
       RET
```

Allen T. Fincher
Suffolk, VA

CON'T FROM PAGE 1 WHAT'S NEW? by Stan Sokolow

telecommunication, personal computers, alternative energy -- all on a 500 acre park. Bill Graham (the rock music promoter), Jim Warren (the West Coast Computer Faire's roller skating guru), the former marketing chief from Lucasfilm (the Star Wars people), a retired Air Force colonel in security and police -- he's collected some crew. It will be a Labor Day weekend extravaganza. Exhibitor booths are $250. Unuson Corporation, 2001 Gateway Place, Suite 500, San Jose, CA 95110, (408) 294-8424.

RockRoy, Inc., introduced DiskSavers colored vinyl floppy diskette sleeves. In quantity 500, 22 cents each, with lower prices for larger quantities. RockRoy, 7721 East Gray Rd, Suite 103, Scottsdale, AZ 85260, (800) 528-2361.

CompuCover makes cloth-backed vinyl dust covers for almost every kind of computer and peripheral, even the Sol ($14.95). CompuCover, P.O. Box 324, Mary Esther, FL 32569, (904) 243-5793.

CPMPOWER is a collection of utility programs for CP/M users being sold by Computing!, 2519 Greenwich St, San Francisco, CA 94123. It will calculate checksums of files, recover erased files, test and mark bad blocks to prevent their use, list file sizes, load file to anywhere in memory, read and write any track/sector, dump memory, fill memory, move blocks of memory, etc. Price $149.

R&P Computer Brokers sells a "used computer blue book" value guide for used mini- and micro-computers. $8.95. 41167 Thurston St, Fremont, CA 94538, (415) 657-9522. They also have a flyer listing computers for sale (commission due when sold).

...ON CONNECTING THE EPROM MX-80 TO THE SOL by R. Shulkin

SIVAD of TEXAS
Roger Shulkin, Distributor

20214 Brondesbury Drive
Katy, Texas 77450

(713) 492-1931
Texas 800 392-2452

May 22, 1982

Dear Stan:

After reading and enjoying PROTEUS for many years, I think that I finally may have something to contribute. I have a SOL with North Star Quad disk drives (2), and presently operate under Release 5.2 of North Star DOS. In my business is a small wholesale type and I use two printers for printing mailing labels and invoices. The printer used for mailing labels recently expired and I replaced it with an Epson MX-80. I thought your readers might be interested in how to connect the Epson to the SOL using the serial port. I use the serial port because I use the parallel port to switch the output between printers under software control using the OUT 253,-command.

To begin, I purchased the Serial Board with the 2K buffer, model 8145, since this was recommended by the local North Star dealer. Installation was easy and took maybe 15 minutes. The switch positions on the serial board and the pin interface to the SOL took a little longer. They are as follows:

	DIP Switch #1:	DIP Switch #2:
1.	On	Off
2.	Off	On
3.	On	Off
4.	Off	On
5.	On	
6.	On	
7.	On	
8.	Off	

Note: Switch #1, 5 & 6 may be changed as required. They determine utilization of the 2K buffer and therefore the speed at which you can dump information to the printer.

Pin Connections:	SOL:	to	Epson MX-80:
	1		1
	2		3
	3		2
	5		4,5,6,8,20
	7		7

I use what I know as the Serial Driver for TI-810, Centronics 761, and IDS 125, 225, consisting of the following fill statements loaded under software control:

```
FILL 51456, 219 \ FILL 51457, 248 \ FILL 51458, 230 \ FILL
51459, 32 \ FILL 51460, 194 \ FILL 51461, 0 \ FILL 51462, 201 \ FILL
51463, 195 \ FILL 51464, 74 \ FILL 51465, 192 \ FILL 51202, 0 \ FILL
51203 , 201 \ GOTO ------
```

The Epson is a great little printer and is a pleasure to use. It does respond to the CHR$(27)+"E" = Enhanced Mode, and to the CHR$(27)+"G" = Double Strike Modes very nicely.

I have been following with great interest your reports of the McVideo upgrade to the 24x80 format. I would like to obtain more information as to any changes required in North Star programs to be sure I could handle any corrections necessary to use the display in my system. I admit to being extremely poor at correcting assembly language programs, and had to get some local help in moving my printer driver programs when I switched from DOS release 5.1 (located at 2A00) to release 5.2 (located at 0E00). Any comments would be greatly appreciated.

Again, many thanks for your excellent articles and valuable information. Please continue!

Roger Shulkin

(Ed. note to Roger Shulkin: I believe that all you need to do to make your existing NorthStar DOS talk to the 24x80 mode of McVideo is just to change the User Area's CALL instructions which use the entry points into Solos. Instead of CALLing an address in the C000 block, they should CALL the corresponding address in the F000 area. For example, instead of calling C004, you call F004. When the video board is switched to 24x80 using the Dual Personality Module, the Sol and Solos go to the F block, and there they know how to act like a 24x80 terminal when output is sent to the Solos output routines. NorthStar owners, let's hear from you.)

UNCLASSIFIED ADS

WANTED

ALS-8 assembly language development system in ROM for Sol. 8080 assembler, editor, simulator on 8 K EPROM board by Processor Tech. Also wanted are the user documents and manual for "Electric Pencil".
Clarise Turner
874 Sunset, Livermore, CA 94550. Evenings call (415)449-4862.

INFORMATION WANTED

Is anyone using a MICRO WORKS DC 80 DIGISECTOR board with the Sol?
M.J. Kerwick, 17 Chapel St., Carrick-On-Suir, Co.Tipperary, IRELAND.

WANTED

Electric Pencil casette that will work with a heathkit printer and Sol serial port.
Lew Pinkham, Delaware Valley Regional High School, RD #1, Frenchtown, NJ 08805

WANTED

Proteus needs a hardware manual for the ALS-8 board,(8K ROM) for the Encyclopedia Processor Technica. If you have one, please send it to us. We will make a copy and return it ASAP.

FOR SALE

Daisy wheel printer terminals -- group purchase -- $675 each, plus shipping (from San Francisco Bay area) or better yet, pick up yourself. These are DTC 300/S KSR terminals, with Hytype I mechanism and keyboard, 30 CPS, RS-232, off-lease units complete with manual, ribbon cartridge, printwheel, 30 day warranty. Adjustable forms tractor available extra. Bryan Devendorf, home (415) 854-2591; work (415) 494-8500.

FOR SALE

MEMORY MERCHANT 16K STATIC RAM -- S-100 -- Bank select, Extended addressing, Segment disable, 4 MHz	$110
GODBOUT 24K STATIC RAM -- S-100 -- 4 MHz	$160
NORTHSTAR 32K DYNAMIC RAM -- S-100 -- Bank select, Self refreshable, 4 MHz	$235
VECTOR GRAPHICS 'FLASHWRITER II' 80x24 video, Keyboard interface, graphics, memory mapped, 256 user definable character set, 4MHz	$160
VECTOR GRAPHICS 64-KEY KEYBOARD -- Numeric Keypad, auto repeat, with case -- 20% discount when purchased with 'FLASHWRITER II'	$100
ALS-8 development package -- Assembler, Text editor, and simulator all with object and source code.	$55
2 back plane boards for the Sol-20	$6 each
2708 personality module board	$12
Walnut sides for the Sol -- set of 2	$10
SOFTWARE TECHNOLOGY music system	$20
CASSETTE RECORDER -- Tape counter, cords, auto-level control	$20
EXTENDED CASETTE BASIC	$25
TREK-80 & VARIOUS GAMES	$10
EXTENDED DISK FORTRAN MANUAL	$3

All items work and are in good condition. PRICES ARE NEGOTIABLE!!

CONTACT: ANDREW BOND, Box 229, GRATON, CA 95444 (707) 823-1232

FOR SALE

Sol 20, 32K Static, Extended Basic, ALS-8, Resident Assem. Checkbook Manager, 8080 Chess and Checkers, all Manuals back issues of Access and Proteus. Make offer. Ed Roberts. 2736 Illinois St. Napa, Ca. 94558 707 226-7557.

E. L. Roberts
2736 Illinois St.
Napa, Ca. 94558

FOR SALE

SOL-20 W/32K IN GOOD WORKING CONDITION. NORTH STAR 5" DRIVE. PLUS REMAINS OF LONG DEFUNCT COMPUTER STORE. 1-P-TECH 2KRO KIT. 1-P-TECH 2KRO ASSEMBLED. 3-P-TECH 16K SEMI KITS. 1-VDM-1 ASSEMBLED. 3-ALS CASSETTES & MANUALS. 2-MUSIC SYSTEM CASSETTES AND MANUALS. 4-EXTENDED BASIC CASSETTES. 1-5K BASIC CASSETTE. 1-MSD VIDEO BOARD 24X80 W/GRAPHICS. 1-VHF CONVERTER. 1-12K BASIC ON CUTS TAPE FOR SOL. OFFER FOR WHOLE PACKAGE PREFERRED, BUT INDIVIDUAL ITEM OFFERS WILL BE CONSIDERED. ALSO HAVE MISC STUFF FOR IMSAI AND CROMEMCO. CALL DENNIS (916) 443-4944, 2322 CAPITOL AVE., SACRAMENTO, CA 95816.

FOR SALE

I have recently upgraded my Sol-20/North Star System from single density disk drives to double density. I have the following surplus material that I would like to offer:

One North Star Single Density Disk Controller Board in good working condition - Price $ 200.00.

Two Shugart 400 Single Density Disk Drives. Drives need to be serviced - Price for the pair $100.00.

Both North Star Single Dnsity Disk Controller and two Shugart 400 Drives per above - Price $275.00. I will include manuals, original North Star-Sol personalized diskette, and North Star Pascal Manuals and original North Star Diskettes (not installed).

Franz Hirner

631 Matsonia Drive
Foster City, California 94404
415-349-3602.

TABLE OF CONTENTS

COMING ATTRACTIONS

In the next issue, we'll have a tutorial article on local area networks, using the Destek network as an example. We've also had requests to talk about applications more. I agree that we need more tutorial articles on applications rather than system problems. I'll do what I can to write some interesting articles, but I could sure use some help from all of you.

Does anyone out there use a spread-sheet calculator program (SuperCalc or the like) extensively? If so, how about a little explanation of how they work and how to use them for novel applications. And what about program generators like Pearl or The Last One? Have you used one? How does it interact with the user to create the desired program? Do you like it?

If you have any other ideas for future articles, let me know, but better yet, write us one. Everyone has something they could share, even the beginners can offer insights for other beginners.

PROTEUS/NEWS

A news journal for owners and users of Processor Technology Corporation computer equipment. Published by Proteus, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483,USA, telephone (415) 368-2300.

Submit items for publication to Proteus News, Attn: Stan Sokolow, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483, USA. Please make submissions as camera-ready as possible by using a fresh, black ribbon and typing single-spaced.

From:
Proteus
1690 Woodside Road, Suite 219
Redwood City, California 94061-3483
USA

FIRST CLASS MAIL

Stephen Maguire
~~SUPO 1092~~0
~~University of Arizona~~
~~8572~~0

2430 N. Dodge Blvd
#Apt E 25
TSN 85716

PROTEUS / NEWS

AN INDEPENDENT NEWSLETTER FOR OWNERS AND USERS OF PROCESSOR TECHNOLOGY CORPORATION COMPUTERS

FORMERLY SOLUS NEWS

1982 - THIRD QUARTER

PUBLISHED BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA

VOL. 5 # 3
SINGLE ISSUE...$7.50 (US)
SINGLE ISSUE...$9.50 (FOREIGN)

LOCAL AREA NETWORKS

A Tutorial

by Stan Sokolow

Local computer networking is an increasingly popular method of providing computer power where it is needed among locations separated by several feet to a few miles. Rather than connecting a bunch of terminals to a large computer, network systems have two or more smaller computers connected to each other. Each computer is called a "node" of the network. A well-designed network will allow users of each node to share the resources of other computers in the network by sending messages to each other. The messages may contain data, programs, commands, inquiries, answers, etc. The messages are also known as packets.

Network topology

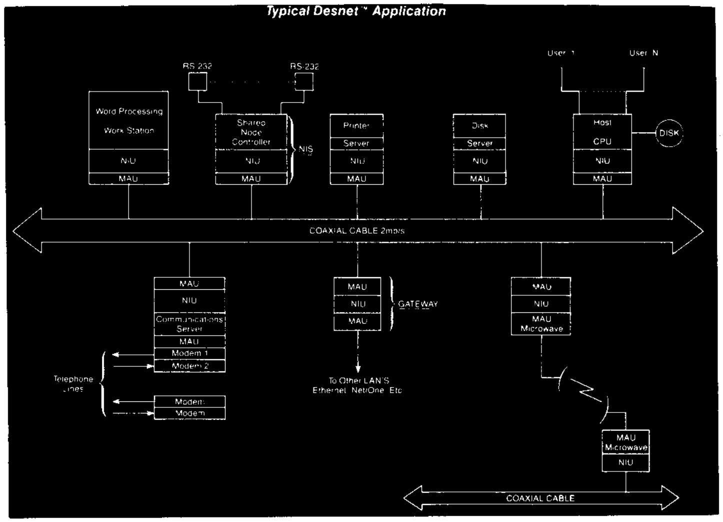
Networks come in several shapes ("topologies"). Some are "star" networks, with one central node and all other nodes attached directly to it. The typical timesharing system with many terminals attached to a central node is an example, but here each terminal is usually not capable of independent computing. Messages from one terminal to the other always pass through the central node, and this is the problem. If the central node malfunctions, no communication can take place between the other nodes.

Another shape is the "ring" network, where each node connects to two adjacent nodes, all arranged in a ring. Messages must be passed from node to node, with each node checking whether the message is for it (in which case it acts upon the message) or for another node (in which event it passes the message on). The problem with the ring is that a failure of one or more nodes can interrupt transmission between some functional nodes.

The most versatile network shape is the "common bus", where all nodes are connected "in parallel" to a common cable (or other data transmission medium). Each node listens to the data on the bus and acts upon messages that are addressed to that node. Messages for another node are simply ignored. If the hardware is designed properly, any node can be off-line or malfunctioning and the remainder of the network can still communicate.

A Real Product

As an example of a common bus topology, lets look at the diagram of a hypothetical application using the Desnet. (Desnet is a product of Destek, manufacturer of computer networking devices, 1923 Landings Drive, Mountain View, CA 94043.)



Typical Desnet™ Application

On the diagram, each node is connected to a coaxial cable which is the common bus. The illustration shows a variety of nodes, but each one connects to the cable in the same fashion, using a Media Access Unit (MAU). The purpose of the MAU is to convert the signals from the node's computer into the form that is required for transmission on the bus.

Transmission Methods

Some networks use "baseband" transmission, where the digital data goes out on the bus (after appropriate amplification). Some use "broadband" transmission, where the data modulates a radio-frequency signal (the "carrier") as it is done in cable tv systems. This allows data and video messages to be moved through the network. And some even use fiber-optic cables, where the data modulates a light signal (solid-state laser) that is carried on a thin strand of glass. This allows an enormous amount of data to be sent concurrently on the same cable, many video and data channels.

The Desnet products are designed to allow the system architect to select the type of MAU most appropriate for the system's needs. It is possible to change the system's transmission method just by installing new MAU's, leaving the other hardware the same.

Between the node computer itself and the MAU is a Network Interface Unit (NIU). This is electrically and logically the link between the computer and the network. For example, the S-100 NIU is an S-100 board that acts as two I/O ports in the host S-100 computer (such as a Sol). One port communicates status and the other carries the data being sent or received. The NIU board is an intelligent device that contains its own Z80 microprocessor and ROM firmware to take care of sending and receiving packets of data, verifying and acknowledging their accuracy using error-detection data in the packets, queing of packets until the host node or destination node is able to receive them, etc.

There are NIU's for a variety of computers, including those using the S-100 bus, the Intel Multibus, the DEC Q-Bus (LSI-11), the Unibus, the Apple bus, the IBM Personal Computer bus, etc. Thus a wide variety of computers can be connected into the same network.

Types of Nodes

Notice that the diagram shows a variety of nodes. Starting at the upper left, the first node is a dedicated word processor, the next is a timeshared computer serving several RS-232 terminals, the third is a "printer server" which consists of a computer dedicated to printing packets on one or more printers connected to it, the fourth is a "file server" consisting of a computer dedicated to management of a disk file storage device, the fifth is another timeshared computer but with its own disk. At the lower left, first is a "communication server" dedicated to handling I/O over telephone lines using modems, the next is a "gateway node" which bridges the gap to another network such as one using a different transmission method, and the last is a microwave communication link to connect with another local network that is some distance away.

This diagram is more complicated than any real application would probably be, but it illustrates the possibilities. For example, a user on the computer in the upper right node could use files located on the disk connected to his own computer, or he could equally easily access files stored on the network file server. All of the users can send data to the printer server node. (The server only allows one user at a time on each printer, of course.)

Small networks

Obviously, in large businesses, such as banks with many branches, the use of networks is essential. But in a single office, it can still make sense to use a local network. Networking several small computers may be a better system for a small business than using a more powerful timeshared computer with several terminals. Can you think of a reason? Well, here are a few.

The timeshared system is more vulnerable to catastrophic failure. If the central computer becomes inoperative, the entire system is down. In a business that depends upon the computer for minute-to-minute operation, this is disastrous. Even if you have a service contract that guarantees four-hour on-site response, you may be without the computer for the whole day while the technician repairs the unit. Can you imagine the chaos and backlog this would create in a business that is all done "online"?

Would it not be far better to have local computing power at each workstation, so that data can still be gathered for later transmission to a malfunctioning node. Some functions of the system would certainly be unavailable, such as files located on a malfunctioning file server, but careful program design will be able to minimize this. It is even possible to have redundant storage, so that two file servers store the same data in the event that one fails.

In a timesharing computer, the more users that the system supports, the more the system response time degrades. Even a fast computer can only be spread so thin before users begin to notice intolerable delays. In systems that supply one processor for each user, such as in local networks, the response remains quick unless there is heavy demand for one of the nodes needed by the user. But even then, the response for locally processed tasks is not degraded. Some tasks can be done asynchronously; that is, while the user is interacting with the node, the NIU can be sending and receiving messages for the next step the user will perform.

In a timesharing computer, increasing the computing power generally means replacing the computer with a faster one, and this generally means a large increment in cost. In a network, more nodes can be added one at a time, generally in smaller increments of cost. As new devices and computers become available, the network can adapt and grow gradually without requiring the old system to be scrapped. For example, a business system made of four Sol's in a network can continue to run the same programs but can be enhanced to include an IBM personal computer as another node. Tasks that the IBM does better than the Sol can be assigned to that node.

Software Makes It Go

Of course, all of the fanciest network hardware is useless without the right programs to make it work. Digital Research (the inventor of CP/M operating system) has developed its own networking software called CP/NET and MP/NET. These allow node computers running CP/M to be connected by anyone's network interface hardware, and to send messages, use files, share printers, etc. Destek is developing their own operating system which is designed from the ground up for network applications.

The International Standards Organization (ISO) has developed a model of network software to facilitate some standardization. They define the network as many layers, or levels of complexity. At the most primitive level is the Physical Link Layer. This is the hardware of the MAU and the cable itself. Functions of this layer are voltage levels, bit-stream timing, data serialization into bits, etc.

The next level is the Data Link Layer. Here message packets are organized, node addresses are recognized, error detection is performed, packets are transmitted or received and acknowledged.

Higher layers of the software are in charge of linking the programs of source and destination nodes, breaking communications and restoring it in another session, and so on.

Complex systems will also provide software to constantly monitor the operation of the network, so that infrequent errors can be detected and repaired before a catastrophic failure occurs.

More to Know

Space does not permit covering more in this issue. There are lots more things to learn about: collision detection, error correction, protocols, device independence, queing, etc.

Perhaps someone will write some more about networks in a future issue.

HELP

Has anybody installed a "Supercalc" 1.12 ver (latest), in a SOL/48 CP/M 2.2 DQ North* 5 1/4" disk system? I have followed all of SORCIM instructions - It BOMBS OUT!!!!

Please call: Millard F. McKinney
43337 Isle Royal
Fremont, CA 94538
(415) 656-7393

Wordcraft

(415) 534-2212

When You Have a Sol and Another Computer

Have you acquired a second computer and retained the Sol? We have our two systems side by side, and the Sol makes a great print buffer.

In our case, we "print" out the serial port of the other system into the Sol. There the text is buffered and sent out the parallel port to the printer.

The program below is a simple buffering routine. Output pauses if you hit the space bar or if a form feed is encountered in the output stream. You toggle the latter feature, basically depending on whether you are printing on continuous forms or stationery. It keeps on buffering input from the serial port during pauses.

By putting a form feed at the end of every document, we can send several accumulated files to the Sol and go on with other work while the Sol pumps out hard copy.

You might want to modify the program to pause on another character, too, distinguishing between form feeds and end of document.

For convenience, we run the program under Micropolis DOS. However, no operating system calls are made, so the program can be re-origined and re-assembled.

We leave the baud rate at 1200; haven't pushed this to its limit, which might depend on your printer timing.

Under MDOS we have a 36 kilobyte print buffer - without buying extra hardware inside or outside the printer.

Whether listing a version of our commercial program, The Micro Link II, or producing a stack of letters, we do our work on the new computer while getting more mileage out of the Sol.

Geoffrey Sinclair

```
*PRTPIPE - in serial port of SOL and out parallel
*Buffer routine
*Contributed by Wordcraft
cout    equ 0C054h ;VDMOT in SOLOS
esc     equ 1Bh
space   equ 20H
cr      equ 0Dh
lf      equ 0Ah
ffeed   equ 0Ch
*
        org 2B00h ;MDOS application area
*
init    in 0F9h ;discard any junk
        call clrbuf
        lxi h,msg
msgout  mov a,m
        ora a
        jz init1
        mov b,a
        call cout
        inx h
        jmp msgout
*
init1   nop
*main loop:
loop    call serchk ;check for input, get it if there
        call prtchk ;give printer a character if ready
        call kybdchk ;let user pause or command
        jmp loop
*
serchk  in 0F8h ;serial status port
        ani 40h ;received data mask
        rz
        in 0F9h ;get data
        lhld inptr
        mov m,a ;buffer the character
        call incr ;increment input pointer
        shld inptr
        ret
*
prtchk  call comptrs ;compare in and out pointers
        rz ;have printed everything in buffer
        in 0FAh ;parallel status
        ani 4 ;printer ready mask, active low
        rnz ;not ready
        lhld outptr
        mov a,m ;get char to print
        out 0FDh ;parallel port
        call incr ;bump the out pointer
        shld outptr
        cpi ffeed
        rnz
        lda pfl ;user can toggle this flag
        ora a
        rz ;pause flag is down
        call halt ;wait for a space
        ret
*
kybdchk call keystat
        rnz ;no key
        call keyin
        cpi space
        cz halt
        ret
*
*Halting action
halt    call serchk ;check input while paused
        call keystat
        jnz halt
        call keyin
        cpi space ;toggle space bar
        rz
        jmp halt
```

```
*
keystat xra a
        in 0FAh ;kybd status
        ani 1
        ret
keyin   in 0FCh
        cpi esc
        jz 04E7h ;exit to MDOS warm start
        cpi 'F'
        cz togfl
        cpi 'C'
        cz clrbuf
        ret
*
*Toggle pause flag:
togfl   push psw
        lda pfl
        cma
        sta pfl
        pop psw
        ret
pfl     db 0FFh ;FF is on, 0 is off
*
*Clear buffer by resetting pointers:
clrbuf  push h
        lxi h,bufstart
        shld inptr
        shld outptr
        pop h
        ret
*
*Increment the pointer in HL
*  At top, circle around to bottom
incr    push psw
        inx h
        mov a,h
        cpi 0C0h ;hi part of C000 upper limit
        jnz incr1
        lxi h,bufstart ;round the top to the bottom
incr1   pop psw
        ret
*
*Compare the in and out pointers
*Set zero flag iff equal
comptrs lhld inptr
        xchg
        lhld outptr
        mov a,h
        sub d
        rnz ;not equal, flag not set
        mov a,l
        sub e
        ret ;flag is correct
*
inptr   ds 2
outptr  ds 2
msg     db cr,lf
        dt 'Printpipe - Passes serial input to the parallel port.'
        db cr,lf
        dt 'Toggle SPACE to pause printing.  Escape exits.'
        db cr,lf
        dt 'While paused, input buffered.'
        db cr,lf
        dt 'Form feed pause is on, toggle with F.'
        db cr,lf
        dt 'Clear buffer with C.'
        db cr,lf,0
bufstart db 0
        end
```

NEVADA FORTRAN for CP/M

Nevada Fortran for CP/M-based systems has been announced by Ellis Computing, a San Francisco software development firm.

The Compiler has been running for 3 years under PTDOS and was converted to the CP /M operating system in April of 1982. Designed specifically for microcomputers, Nevada Fortran is powerful, yet easy to use. It's both a subset and superset of ANSI 1966 Fortran. Popular extensions include: IF-THEN-ELSE constructs, TRACE style debudgging, COPY statement, Arrays up to 7 dimensions and Random Access file support.

The high performance compiler generates 8080 machine language on all CP/M based ystems with at least 32K RAM. Dynamic object module loading and chaining takes place in seconds using the same fast loader as the firms Nevada COBOL Compiler.

COBOL, FORTRAN, PILOT AND EDIT PRICE REDUCTIONS!

Ellis Computing, a San Francisco based software development firm, announces the immediate price reduction of their popular CP/M-based software series. Nevada COBOL, Nevada FORTRAN, Nevada PILOT, and Nevada EDIT have all been reduced to $29.95 each. These are the same professional packages that have sold for as much as $300.00 per copy.

According to Chuck Ellis, it's time the micro-users' were given an affordable alternative to BASIC. By providing our packages at these reasonable prices, we will be fulfilling that need.

<Excerpt from a stock-market news item on Softech Microsystems>

IBM SELECTS SOFTECH'S USCD-P

IBM has selected SofTech Microsystem's USCD-p operating system as standard for data processing applications on its "Displaywriter" word processor. Royalty income will ensue, but it is hard for us to estimate how much and when.

The announcement is far more important for the prestige it brings. Up to now, the USCD-p system has been licensed to most of the important microcomputer manufacturers, but has been sold only as an alternative (the IBM microcomputer) or as a separately priced option. CP/M, on the other hand, has become a standard for most of the new micros. Therefore, despite the fact that USCD-p has not also been used, most notably in Softwre Publishing's "PFS", probably the best selling data base management program for micros. The IBM announcement will make microcomputer manufacturers and software developers think twice and could arrest the CP/M trend.

AN INTRODUCTORY DEAL ON REMOVABLE MEDIA WINCHESTER DISK

Carter Collins has discovered a mini-Winchester disk that has an "industry standard" removable disk pack, 5 megabytes capacity, for less than $500 in quantity two. He would like to find someone else interested in buying one with him. This is apparently a one-time offer to introduce the product. It is the bare drive and will need a controller compatible with Seagate Technology standard interface used by this new product, but Xebec sells one for $295. Sounds like a good deal for someone who can handle the details of installing it. If interested, contact Carter Collins at (415) 561-1633 DAYS

Dear Stan:

Enclosed with this letter is documentation of ABLE, a line editor written for the Sol-20. ABLE requires at minimum a 64-column screen 48K Sol-20 with at least one CP/M-based disk. It's at its best with the 80-column screen upgrade and 58K RAM system, such as offered by Micro-Complex.

ABLE is meant to be a companion editor to Microsoft's BASIC-80 interpreter and compiler, as well as an all-around editor for processing high level language source files. ABLE is, in fact, one of the few editors capable of editing line-free BASIC-80 source files for the compiler, and providing logical line continuation in conjunction with a visible physical line break. Such a feature is needed to exploit the ability of the compiler to accept physically long statements, as well as pretty-printed text in IF-THEN or AND/OR/XOR constructs.

The syntax of ABLE is very close to that of the BASIC-80 editor, and so may be learned very quickly. If you like what ABLE can do, I'd like to sell ABLE to Proteus members at a low price.

Yours very truly,

Daniel S. Hunt

If any Proteus members are interested please contact Dan Hunt directly:
Daniel S. Hunt
359 Princeton Drive
Costa Mesa, CA 92626
(home phone (619)549-8673)

```
*************** DOCUMENTATION FOR "ABLE" ****************
*                                                       *
*             Rev. 1.7      MAY 7, 1982                 *
*              (C) 1982, Daniel S. Hunt                 *
*               GREEN VALLEY SOFTWARE                   *
*               359 PRINCETON DRIVE                     *
*               COSTA MESA, CA 92626                    *
*               (714) 549-8673                          *
*                                                       *
*********************************************************
```

ABLE is a line editor with characteristics similar to Microsoft's EDIT-80* and UCSD's Pascal system Yaloe. But its features are not as complex. So ABLE is easy to use. It is designed to edit high-level language files in the 200- to 1300-line range, on CRT or hard copy terminals.

ABLE is a particularly suitable companion editor for BASIC-80*. Both the BASIC-80 interpreter and compiler require unusual end-of-line treatment to extend logical statements across physical lines. ABLE's ability to do this makes it possihle to apply line breaks and indentation in mid-statement to add structure to BASIC source code.

ABLE's edit mode syntax is like BASIC-80's. Command mode features include essentials not found in BASIC-80: global search and replace, partial file saves (and merges), buffering of killed lines, and the ability to move and copy individual lines.

ABLE's text display is friendly by line editor standards. The context of a line is easily perceived with ABLE's variable and automatic multi-line paging commands, which scroll up or down in a file. In addition, you may move up or down a single line at a time with a single key entry.

Single key commands, T and B, give the user immediate access to the top or bottom of a text file. No hidden toggles are used (example: the page down (@<CR>) key is different than the page up (^<CR>) key.

Editor status may be queried to reveal the active file name,total lines in the file, memory remaining and the contents of the line saved in the copy buffer.

Line insertion may precede or follow the current line. The latter option requires no line number -- to facilitate fast entry of new lines into the middle of an existing text.

File management routines include file load, file save, file merge, file name reset and ability to view disk directories from within ABLE. File names may be activated directly from CP/M as a command argument, or reset from within ABLE. File backup is left to the option of the user, to conserve disk space. Portions of the working text may be saved, a most useful feature for extracting potential library routines from large programs or for temporarily storing large blocks of text that can then be merged back into a different location.

What are ABLE's disadvantages compared to the current crop of screen editors and word processors? It works in memory only, and it is a line-editor, which takes some getting used to.

In-memory editors trade off maximum file size, limited by RAM. But they gain speed of access to any part of the text file. Line editors don't have random cursor action, but are handier with compilers for correcting errors, which are indicated by line number. If your editing problem requires viewing two or three parts of an edited file at one time, a line editor offers the simplest and least expensive way to do so.

COMMAND SUMMARY

MODES -- ABLE is invoked from CP/M by typing "ABLE". An optional file name may be used in the command, as in: ABLE MYPROG.EXT. If the option is used, the text MYPROG.EXT will be loaded automatically as ABLE boots up. If the optional filename is not used, no file is loaded. You should assign a file name to the new text by using the FR command.

ABLE boots up in the command mode. This mode allows you to access different parts of the text, load, save or merge files, display single or grouped lines, insert, kill, move, copy or append lines, and query the editor's status. The edit mode is invoked from the command mode. On the next page is a schematic summarizing instructions and the groups of action to which they apply.

```
                                  CP/M
                                    |
                              ABLE [MYPROG.EXT]
                                    |
                            ABLE COMMAND MODE KEYS


(En) (FS,FL,FM,FA,FB,FP,FR) (T,B,D,L,@,^,-,?,V,!) (A,K,I,S,R,Q,M,C,P)

  |                  |                 |                |
  |                  |                 |                |
EDIT MODE KEYS     FILE          TEXT DISPLAY &     OTHER
(X,H,I,C,D,L,S,Q,A)  ACCESS      STATUS INQUIRY     ACTIONS
                                                        |
                                                        |
                                                        |
                                                        |
                                                        |
                                                        |
           KILL, COPY, MOVE, APPEND, INSERT, SEARCH, REPLACE
```

ABLE COMMANDS IN DETAIL

ABLE Prompts: Two types of prompts appear on the terminal to help the user distinguish between command mode and edit mode. The edit mode prompt is preceded by a line number. Line numbers are sequential from 1 to 1300. Lines do not belong to a specific line number and may change if additions or deletions are made to preceding lines.

Entries of commands in command mode must all be followed by a carriage return (shown as <CR> in this manual). To edit a command line, use CP/M syntax -- i.e., CTRL-U to cancel a line, CTRL-R to redisplay a line, etc. In edit mode the commands do not have to be followed by <CR> with the sole exception of the <CR> required to leave edit mode.

Note that line display begins with tab position 9 on the screen. This is to provide separation between the line number and the first character of the text line. As the BASIC-80 runtime module in ABLE "remembers" line position, and defaults to an 8-space tab, the tab 9 positioning of the first character in the line allows tabulation to be displayed exactly as it will be printed. A line may extended beyond the end of the screen, and will appear to wrap around to the first position on the next line.

Examples of prompts are:

? -- You are in command mode.

23: -- You are in edit mode.

COMMANDS FOR FILE ACCESS

File commands are two letter sequences and they all begin with "F" (for File).

FL (File Load) -- Loads a CP/M format text file into the work area for editing. No filename is needed after this this command. The name established in the CP/M command argument is used, or the name most recently changed with the FR (file name reset) command. If the file is not found, a "File error" message is displayed.

FS (File Save) -- Saves the text in the work area to the disk. No filename is used after this command. FS uses the file name used during program invocation, or the name most recently established with the FR command. This means that you may quickly and repeatedly make temporary saves of the text as you change or add to it. This practice is suggested to prevent loss of text from power surges or memory failures.

FP (Partial File Save) -- Saves a part of the working text to disk. After typing FP, you are prompted for a filename, and for the first and last line numbers of the block to be saved. The file name must be different from the active file name, which continues to be active after the FP operation.
FR (File name Reset) -- Changes the active file name. This feature allows you to choose a new active file name for loading or loading, AFTER you have invoked ABLE. Type FM. On the prompt, enter legal CP/M file name, either in upper or lower case. The name will automatically be converted to upper case.

FM (File Merge) -- Merges a file to any place in the work area. After the command, FM, you'll be asked for the merge file name. Then you'll be asked in front of which line number the file is to be merged. EXAMPLE: If you indicate line #6, then the text beginning at line #6 will move down to make room for the new file. If the merged file is not found, a "File error" message will be displayed. The merge file name does not change the active file name.

FA (View directory on A), FB (View directory on B) -- These are the only commands needed to display the file names on either of two disks. If your system has more than 2 disks, provision for extra disk names may be added to the file management subroutine. BEWARE! Query to the directory of an unoccupied disk unit will crash the editor.

TEXT DISPLAY, TEXT LOCATION AND STATUS INQUIRY

Once you have loaded a file, text may be displayed one line or 16 lines at a time, forward or backward in the file.

<CR> -- Displays the NEXT relative line. On loading a file, the current line is line #0. Typing <CR> will display line #1.

L[line number]<CR> -- Alone, L shows the current line. If followed immediately by a number, L will display the line at that line number.

D[line number]<CR> -- Displays a group of lines, the last of which is the current line. Use of the optional line number resets the current line and thus the last line of the display. The number of lines displayed is set with the V command.

@<C~R> -- Displays the next line group down in the file, relatively from the current line before the command was entered. The V command sets the size of the line group.

^<CR> -- Displays prior line group, relatively to the current line. Group size is set with the V command.

V<number of lines><CR> -- Sets the number of lines displayed in a group display command E.g., D,@,or ^ commands will display a 5-line group after the command V5. This command is very useful to display two or more narrow groups of text adjacently on the screen.
T<CR> -- Moves you to the Top of the text area (line 0). Note: Line 0 never contains text. So D,L do not reveal anything until you've pressed CR or @<CR> to advance the display into the text area.

B<CR> — B is for Bottom. Moves you to the last line in the text.

?[n]<CR> — Displays current file name, line number, total lines in the file, number of characters left unused in memory and the line buffered by Move, Copy or Kill. The optional number will reset the current line number.

APPEND, MOVE, INSERT OR KILL TEXT

Appending or inserting a line into existing text creates or vacates a line number for the user, and automatically puts the user in EDIT mode, in the line extend (X) condition.

A<CR> — Append a line to the end of the file. If the last line of the file is 99, then A<CR> will create line #100 and prompt the user with the edit prompt. You need only to type in new characters and type <CR> when finished. Edit mode controls apply while in the act of adding a line. "A" always moves the current line number to the first position following the last line of the file.

I[line number]<CR> — If the optional line number follows the command, I, then you are inserting the line AHEAD of the line number shown in the command. If you enter I alone, then you are inserting a new line AFTER the current line: This is done for speed. You may rapidly insert several lines of text into the middle of existing copy, without having to bother to enter line numbers. Nor do you have to worry about overwriting the lines that follow. If line insertion occurs at the beginning of a large file, a noticeable pause occurs as the following lines are moved to make space for the new line.

CONTROL CHARACTER INSERTION — Control characters are recognized from the keyboard and may be embedded in text lines. This is most useful for the insertion of formfeeds at desired page jumps in the text. As some terminals consider control characters invisible, it is suggested that a separate line be used for their insertion. Example: To insert a formfeed, type i<CR> at the line desired. When you get the empty line and edit mode prompt, type CTRL-L (formfeed) followed by a carriage return to end the line. A clear screen character (CTRL-K on the Sol-20) may be inserted in the same fashion. You may use CP/M's PIP with the F extension to remove formfeeds from the text file by copying it into a new file. Example: PIP FILETWO = FILEONE[F].

K[line number]<CR> — If used alone, "K" will kill a line at the current line number and the text will be closed up to fill the gap. So be sure you know what that current line number is! (Display it with L). If you include the optional line number, "K" kills the line at the number indicated. Once you've killed the line, the line which now fills that space is displayed. If you want to kill it, just bang in another "K". ABLE always buffers the very last line killed, by the way. If you want the line restored, type "P" and the target line number.

!<CR> — Creates a new text area ready for use. Blanks all existing text. Be sure you save any needed text before using the ! command.

M[n] — Move a line. [n] is the current location of the line. The line is vacated and closed up. ABLE buffers the line, so you can wander around the file to find a place to put it, with the command "P[n]".

C[n] — Copy a line. The same as M[n] except that the line is copied, not removed from its original place. Type C[n] to buffer the line, type P[n] to put a copy of the line in the location you choose. Rapid replication of a text line may be achieved by alternately typing C[n] and P (no number).

P[n] — Place a line buffered by M or C commands. "P" alone means put it after the current line displayed. P[n] means put it at the line indicated by [n].

ENTERING AND USING EDIT MODE

Edit mode has its own commands. The first initial stands for what they do, like s for search, x for extend, h for hack, and so on. <CR> gets you out of edit and back to command mode. Enter edit mode by typing:

E[line number]<CR> — "E" alone means edit the current line. This is handy if you've gone back to command mode only to discover that the current line still needs fixing. Just type "E" to get back to edit mode and fix the line. Used with the optional line number, "E" enters edit mode at the line indicated.

EDIT MODE COMMANDS:

Space Bar — Move the cursor to the right one space. Nothing is done to the text; it merely appears from under the cursor. To speed things up, you may simultaneously press the space bar and the repeat key if you have one.

D — Delete the character under the cursor. After you type "D", the deleted character appears surrounded by two reverse slashes, as in "bud\d\get".

K<character> — Delete all characters from the cursor up to but not including the character typed after "K". The deleted area is surrounded by reverse slashmarks, as in "The \quick\ brown fox".

S<character> — Searches for the next occurrence of the character indicated and moves the cursor there.

L — List the edit line with all modifications up to this moment.

A — Disregard all changes made since you began editing the line and restore the line to its original condition.

ESCAPE — The following commands, I,X,H,C, are cancelled by use of the escape key.

DEL — The DEL (delete) key may be used to erase characters when the commands I,X,H,C are in effect.

I — Insert one or more characters beginning at the current cursor position. To disable insertion, touch the ESCAPE key (or leave edit mode by using <CR>). You may delete characters added merely by touching the DEL key; the cursor will move backwards and the characters will be rubbed out.

X — Extend the current line, i.e., allows characters to be added at the end of the line. Touch the ESCAPE key to disable extension. DEL key deletes characters one by one from right end of the line.

H — Hack off all characters at the cursor and to the right and allow new ones to be added on. ESCAPE disables this condition. The DEL key may be used to delete to the left from the point at which the H mode has been entered.

C — Change one or more characters, beginning with the one at the current cursor position. Type "C". The next character typed will operate on the character under the cursor and change it. To stop changing characters, type ESCAPE, or <CR> to exit from edit mode. The change function differs from the Microsoft convention in that you don't have to count the length of the change to be accomplished.

EDITING BASIC-80 WITH ABLE

ABLE -- written in Microsoft* BASIC -- will edit a Microsoft BASIC source file created with the interpreter. But first it must be saved in ASCII mode from the interpreter by using the special BASIC-80 extension after the SAVE command as in:

```
SAVE "FILENAME.BAS",A.
```

The ABLE user may break a long BASIC-80 logical line into several physical lines by A) causing physical-end-of-line with a line feed and tab (in which case the required following carriage return is automatically generated and echoed), or B) by embedding an underline character at the end of the physical line and closing the line with a carriage return. Method A) causes the logical line to stay within one ABLE physical line, which has a maximum length of 255 characters, offering more room than the 128-byte limit of the BASIC-80 compiler.

Method B), the underline <CR> termination, may be used if logical line lengths are longer than 255 characters. Method A) will be readily visible to the programmer, while Method B) may be hidden on some CRT's. Line breaking permits BASIC-80 compiler users to use the /C option, in which all line numbers except those referenced by GOTO and GOSUB may be omitted from BASIC-80 for better readability.

SAMPLE LINE STRIPPING PROGRAM IN BASIC

The program below -- which has been edited in the interpreter and prettied up to a small degree with indentation -- can be cleaned up in a striking manner, if it is intended for the compiler. In using the BASIC-80 interpreter and compiler in tandem, it is usual to create and iteratively debug the BASIC program until its size becomes unwieldy, or until the standard library of routines is operating in a satisfactory manner. Then you can switch to a text editor, and test the results in the compiler. Before you do the latter, it would be nice to eliminate all but the essential line numbers.

That's what this program does. You're free to make use of it for non-commercial purposes. Save the BASIC source file in "A" mode, then run it through the line stripper.

```
10 REM BASIC-80 LINE NUMBER STRIPPER
20 REM (C) 1982 Daniel S. Hunt     (714)549-8673
30 '
40 PRINT"BASIC SOURCE LINE NUMBER STRIPPER" : PRINT
50 INPUT"SOURCE FILE";FA$
60 INPUT"DESTINATION FILE";FB$
70 OPEN "I",1,FA$
80 OPEN "O",2,FB$
90 '
100 WHILE -1
110     LINE INPUT #1,A$
120     IF EOF(1) THEN 180
130     GOSUB 210 :'STRIP NUMBER
140     PRINT #2,A$
150 WEND
160 '
170 REM STRIP LAST LINE AND ADD ^Z MARK FOR CP/M EOF MARKER
180 GOSUB 210:PRINT #2,A$:PRINT #2,CHR$(26) : CLOSE
190 END
200 '
210 REM SUBROUTINE STRIP LEADING NUMERICS
220 FOR IDX = 1 TO LEN(A$)
230     CH$ = MID$(A$,IDX,1)
240     IF CH$ < "0" OR CH$ > "9" THEN 260
250 NEXT
260 A$ = RIGHT$(A$,LEN(A$)-(IDX))
270 RETURN
```

THE SAME PROGRAM "STRIPPED, PIPPED AND ABLED"

The output from the line stripping program appears below. For an extra touch of elegance, the CP/M PIP utility's [L] option was used to copy the output of the line stripper into yet another file with all characters converted to lower case. The command to do this would be: PIP LCASEPRG.BAS=STRIPPED.BAS[L]...

```
rem basic-80 line number stripper
rem (c) 1982 daniel s. hunt     (714)549-8673
'
print"BASIC-80 source line number stripper" : print
input"source file";fa$
input"destination file";fb$
open "i",1,fa$
open "o",2,fb$
'
while -1
        line input #1,a$
        if eof(1) then 180
        gosub 210 :'strip number
        print #2,a$
wend
'
180 rem strip last line and add ^z mark for cp/m eof marker
gosub 210 : print #2,a$ : print #2,chr$(26) : close
end
'
210 rem subroutine to strip leading numerics
for idx = 1 to len(a$)
        ch$ = mid$(a$,idx,1)
        if ch$ < "0"
        or ch$ > "9"
                then 260
next

260  a$ = right$(a$,len(a$)-(idx))
return
```

Note that the IF-OR-THEN construct just above line 260 in the stripped, lower-case version of the program was edited with ABLE's special logical line continuation sequence. At the end of the physical line a line feed is entered, followed by tabs and spaces for the desired indentation. ABLE automatically inserts a carriage return following the line feed (instead of preceding it) and the BASIC-80 interpreter or compiler will be satisfied that it may continue the logical line.

ABLE'S EDITING ALGORITHM --- Editing a line involves splitting an edit string into two stacks. This makes for relatively simple procedures for modifying lines. It also permits the programmer to avoid the use of line length counters. On entry, the left stack is BUFF$. It is empty. As the cursor moves right in the file it looks at the topmost (leftmost) character in RBUF$, does whatever is needed in the way of adding, inserting, changing or deleting it, and pulls it off the RBUF$ stack with a LEFT$ function. If the character survives scrutiny it is placed on the top (right) of BUFF$ stack, which will continue to grow as the cursor is move to the right. At the end of the line, the right stack will be reestablished as the cursor is move back to the beginning of the line. This approach is much simpler than trying to keep track of the cursor with position pointers. With pointers you can get disoriented and write excessive amounts of code. Such is the wonder of BASIC string handling that stack functions can be represented so simple a manner.

Due to BASIC's method of garbage collection, ABLE exhibits a few strange quirks. One of these is an occasional pause as ABLE executes the S command in edit mode. The pause occurs as BASIC-80 cleans up unused string space. Length of the pause is hardly noticeable in the compiled version of ABLE, but can last a few seconds in the interpreted version.

ABLE'S STATISTICS

ABLE.COM file size: 24K+
Maximum line length: 255
Language: Native code BASIC-80*

Suggested minimum system: 48K CP/M* (Edits 18K file)
Best system: 58K+ CP/M (Edits 26K+ file)

Configuration of this demo:

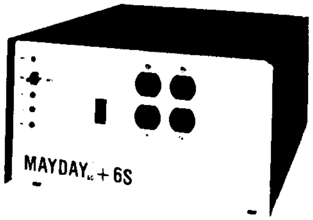
    Clear screen character: CHR$(11) (Sol-20/VDM)
    Delete character: CHR$(127) (Sol-20/VDM)
    Rubout character: CHR$(1) + CHR$(32) + CHR$(1) (Sol-20)
    Maximum lines: 1300
    Edit columns on 80-column screen: 72
    Edit columns on 64-column screen: 56

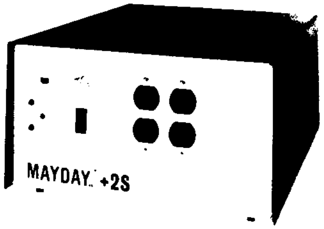
* CP/M is a registered trademark of Digital Research.
* BASIC-80, EDIT-80 are registered trademarks of Microsoft.

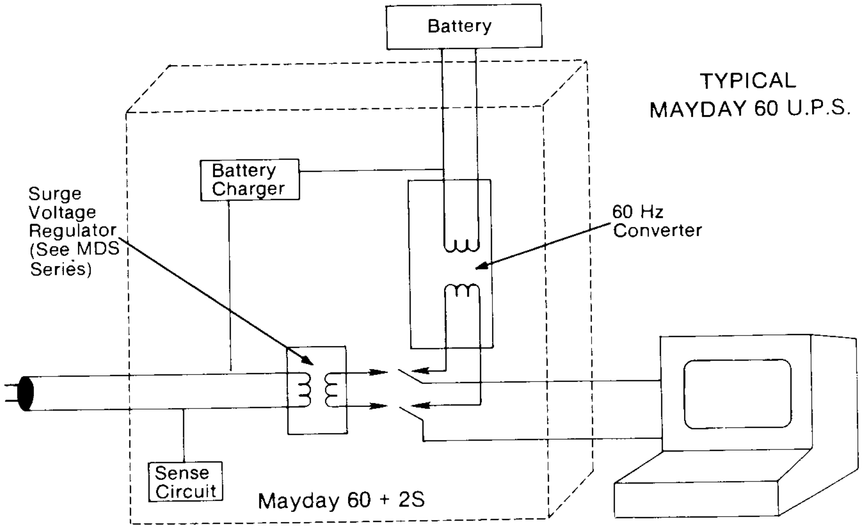
Daniel S. Hunt
Green Valley Software
359 Princeton Drive
Costa Mesa, CA 92626
(home phone (619)549-8673)

TIME FOR YOUR LIST FOR SANTA CLAUS

In the next issue, I would like to publish some of our collective ideas for Christmas wishes, like "I'd like someone to make a ... for my Sol" or "I wish someone would show me how to ... with my Sol." For example, ever wanted the computer to help you minimize utility bills by managing window shades, zone heating, lawn sprinkling, etc.? Maybe there is something like this that you would like to do but don't know how. Send your wildest ideas to Santa, care of Proteus News, and I'll publish the best in the next issue. If you wish hard enough, Santa may bring it to you in a future issue.







TYPICAL MAYDAY 60 U.P.S.

Mayday 60 Specifications

Output Voltage (on D.C.)—Regulated 120 VRMS + 3 volts over 80% of battery energy. Harmonic distortion is less than 5%, typically 1%, 60 cycle crystal frequency controlled to .01%.

Short Circuit Protected—When on DC, by ferro-resonant transformer, when on AC only with surge voltage regulation.

Switchover Rate—Typically less than 8ms, maximum is less than 12ms.

Recharge Time—Four to six hours for 90% charge on battery.

60Hz Mayday Specifications

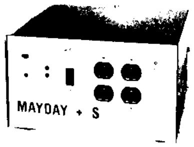
MODEL	(VA) WATTS	LINE SURGE PROTECTION	DIMENSIONS (INCHES)	WEIGHT	BATTERIES (Included with system)
M60	150	NO	11 x 12 x 7	23 lbs.	1
M60 + S	150	YES	14 x 16 x 8	34 lbs.	1
M60 + 2	250	NO	14 x 16 x 8	38 lbs.	1
M60 + 2S	250	YES	14 x 16 x 8	52 lbs.	1
M60 + 6	600	NO	14 x 16 x 8	42 lbs.	2
M60 + 6S	600	YES	14 x 16 x 8	63 lbs.	2
M60 + 10*	1000	NO	17 x 21 x 9	90 lbs.	4

*Switch-over to D.C. exceeds one cycle.

Standard batteries for the above systems are Deep Cycle, 12 Volt Lead Acid, 30 Amp Hour in separate cases.
Non-liquid lead acid batteries optional.
All above systems operate on 115 Volts A.C., 60Hz input/60Hz output.
Conversion to 220 volts/50Hz Sine wave output available at extra cost.

All units F.O.B. New Durham, New Hampshire, U.S.A.

MAYDAY™



MAYDAY60 SERIES
60Hz SINE WAVE OUTPUT

for mini/micro-computers and micro processors that require a 60Hz sine wave output with power requirements of 150 to 1000 watts. Battery pack included.

MAYDAY60	150 watts	$ 400.00
MAYDAY60+S	150 watts	475.00
MAYDAY60+2	250 watts	525.00
MAYDAY60+2S	250 watts	700.00
MAYDAY60+6	600 watts	900.00
MAYDAY60+6S	600 watts	1140.00
MAYDAY60+10*	1000 watts	1950.00

*Switch-over time 50 ms

"S" designator denotes surge protection built in.

MAYDAY60+C

Converter: Non-Switching

Designed for situations where a non-switching backup power supply is required. These units control frequency to ± 0.01% with crystal oscillator control. Battery not included.

			Battery pack required
MAYDAY60+2C	250 watts	800.00	A
MAYDAY60+6C	600 watts	1325.00	B
MAYDAY60+10C	1000 watts	2295.00	C

MAYDAY for 220 volt 50/60Hz operations are available upon request.

Prices effective March 10, 1982

MDS SERIES

Line Surge Protectors
Ferro-Resonant Voltage Regulator

MDS-150	150 watts	$ 90.00
MDS-250	250 watts	185.00
MDS-600	600 watts	275.00
MDS-1000	1000 watts	500.00

120Hz SQUARE WAVE OUTPUT

For TRS-80 Model I and III, Atari, HP-85, PET and any microprocessing equipment utilizing 120Hz square wave output. Battery pack included.

MINI-MAYDAY	50 watts	$145.00
MAYDAY	150 watts	240.00
MAYDAY+S	150 watts	325.00
MAYDAY+2	250 watts	350.00
MAYDAY+2S	250 watts	500.00

"S" designator denotes surge protection built in.

Optional "SMART" Circuit for 120Hz Units $60.00.

BATTERY PACK

A	12 volts	$ 50.00
B	24 volts	100.00
C	48 volts	200.00
D	72 volts	300.00

Rack mounted metal case available.
Non liquid lead acid battery available.

Prices applicable when purchased with unit

Total harmonic distortion typically 1% in all sine wave units. All units F.O.B. New Durham, N.H. Prices subject to change without notice.

SUN RESEARCH INC.
Box 210 Old Bay Road New Durham, N.H. 03855
(603) 859-7110 TWX 510-297-4444

A NEW PROTEUS ITEM --- P 27

$30.00 ON HELIOS DISK

====>> SERVICE TESTS USERS GUIDE <<====

The service test disk contains 10 programs intended to aid the service technician test Sols, memories and Helios systems. The tests are all "Image type" files and may be loaded into memory by simply typing the desired test name and a cariage return. The following is a brief summary of the available tests:

16KRA: not used

Tests a 16K block of contiguous memory beginning at 1000 to 4FFF continuously until the escape key is depressed. The program loads at 0000 and is approximately 256 bytes long. When executed, the program will not display anything on screen until first pass has completed. Details of this test are at the end of the 16KRA manual.

32KRA:

Tests a 32K block of contiguous memory at location 0000 - 7FFF in either a single pass or a continuous loop mode. The test loads at memory location C900 (system RAM) and is approximately 256 bytes long. The test begins by allowing the option of continuous or single pass and prints out a pattern of "G" and/or "X" as in the 16KRA test

48K:

This test will check 48K of contiguous memory beginning at 0000 - BFFF in either the continuous or single pass mode. The test loads at C900 and is approximately 256 bytes long. Typing "C" in the beginning sets the continuous mode which will run until the "escape" key is depressed. The test will print a pattern of "G" or "X" as in all the other memory tests.

DMARD:

This test will do DMA reads from the memory onto the disk and uses PTDOS. The program loads at BCC0 and is approximately 400 bytes long. The memory from which the test reads is determined by the first and second parameters when typing the test name from PTDOS. For example:

```
DMARD 0,7   <CR>
```

will do a DMA read from 0000 to 7FFF in 4K blocks to the disk. Each 4K block of memory is read 9 times. A bad read will be indicated by an error message and the user has the option in the beginning to either stop on an error or continue on an error. The test will terminate when the "Mode Select" key is depressed.

DMAWR:

This test will do DMA writes into memory from the disk using PTDOS. The program loads at BCC0 and is approximately 400 bytes long. The memory which will be tested is specified by the parameters used when typing the file name from PTDOS. For example:

```
DMAWR 0, 7
```

will do a DMA write into memory locations 0000 - 7FFF in 4K increments writing to each block a total of 9 times. The test will either stop on eroors or continue on errors but in either case will report the failure. Pressing the "Mode Select" key will return the user to PTDOS.

SOLT:

This test loads at memory location E000 and is approximately 5K long. With this test it is possible to verify the correct operation of the Sol ports. Various tests are available from the menu which comes up when the test is executed by typing the appropriate letter.

DOST:

This test is an example of a PTDOS "DO Macro" which is used to test the Sol System III and the PTDOS disk commands. The test can be written into the START.UP file where it will execute continuously. By editing the file, the user can observe the different commands being used. The other way to use this test in single pass mode is to type "DO DOST". The only errors reported will be PTDOS errors since this test runs under PTDOS control.

DISKT:

This program is the DISK diagnaostic program detailed in the Service and Maintenance Manual and the HELIOS manual. The test is located from 0000 - 2FFF and uses 3000 - 3FFF as its buffer. The program is execuated at location 0003.

DISKCHK:

This test is a PTDOS command file. By typing the name "DISKCHK" PTDOS compares every file on the disk to the directory to insure the file structures are correct. This test does not say the Data within the file is correct but the file integrity is OK. Any errors here will be reported as PTDOS errors. This program will give memory overflow errors on large programs if the buffer is not set low enough. It is a quick way to check whether a disk is bad or good and is not really a diagnostic tool.

SIMU:

This program is to be used in conjunction with the Alignment section of the Service and Maintenance manual. The program loads at 0000 - 1FFF and executes at 0. Basically the program is meant to exercise the drive hardware during disk alignment and simulates in software the PERSCI drive exerciser. SIMU command mode is designated by an asterisk and the "ESCAPE" key returns the user to SOLOS.

Dear Stan:

I wrote to you several months ago about the Electric Pencil (possibility PROTEUS reviving it, etc.), now that I see ads in the magazines for TRS Electric Pencil II, I understand why it couldn't be done. (Obviously someone else holds the rights to Pencil, and it appears that they either not realize that there is still a market for non-TRS-80 versions, or they don't want to bother supporting them.) In any case, I am still using my Electric Pencil I Version SS. I have interfaced Pencil to both North Star DOS and CP/M, so that the cassette I/O commands do disk I/O instead. The necessary modifications consist of relocating Pencil, and entering some code to emulate SOLOS tape I/O routines with disk I/O. I can make the entire thing available to anyone who is interested; shall I send a copy to you for PROTEUS?

I received a reply (from your secretary, I assume) stating that my name has been added to Bob Marsh's list of people interested in getting a better S-100 expansion backplane for the SOL. Do you know if anything is happening with that? I will probably want two boards. That also brings up another idea that I've had: By plugging in a similar five-slot backplane, but with only power supply bus lines connected from the main board to the upper few slots, a SOL owner would be able to plug in one of the new, (relatively) inexpensive and low-power single-board S-100 computers, and run that as the main CPU with the SOL serving as a smart terminal (and possibly an I/O front end, for things like disk I/O.) The single-board computer I have in mind is the Advanced Micro Digital board with a Z-80A, 64K memory, double density disk controller, two serial and two parallel I/O ports, up to 4K shadow EPROM, extended addressing, and a real time interrupt clock, all for $800 to $900. Of course, there are others (even 16-bit CPU's) that could be used as well. Has anyone tried doing this, or does anyone know of any reason why this can't be done? It sure does seem like an attractive possibility.

Finally, I would like to commend you on the continued high quality of PROTEUS/NEWS, and to thank you for your efforts on behalf of all the SOL owners around the world. The fact that we still use our SOL's after all this time attests to the basic quality of the product. It's really a shame that the full potential was never realized!

Yours truly,

John Osudar

John Osudar

P. O. Box 1451
Homewood, IL 60430
September 12, 1982

(Editor's reply to John Osudar:)

Dear John,

Please DO send us an article on your modifications to Electric Pencil. Lots of people still feel most comfortable with that as their word processor, and I'm sure some would like to get the Pencil up on their disk system.

About the backplane board. We are looking into price quotations for a small quantity. We would like to have a larger quantity to get the price down, but so far we only have about 25 boards' interest from members. If any readers are interested in buying an extra backplane, please let us know soon. For those who have forgotten what this is all about, let me remind you: The Sol's S-100 backplane board has edge-connectors which tend to become loose and unreliable after repeated use. Bob Marsh has artwork to make new board, so people can replace the board when it goes bad. (Symptoms: intermittent hardware errors of unknown origin, especially errors that are sensitive to movement of the computer itself.) Price estimate: $25 to $40 depending upon how many of you want to buy one. Remember, this may be our only chance to get this kind of spare part!

I have several questions regarding the SOL.

1. I am interested in increasing the clock speed from 2MHz to 4MHz.

2. I also would like to upgrade to a Z80 processor. It appears that more and more software is being developed for the Z80.

I would appreciate any information regarding the above.

Sincerely,

Michael C. Wojnar
Michael C. Wojnar

28 Walden Lane South
Burnsville Mn 55337
October 5, 1982

EDITOR' NOTE:
We have a Z-80 upgrade module in finale development stages. It will run at 2 or 4 MHZ (even 6 MHZ if desired). We'll announce it in Proteus News when its ready.

October 5, 1982

Dear Stan,

Just a quick note in answer to John Barber's letter in Proteus Vol 5/2:

If Wordstar has been installed to use memory mapped video on the Sol, the address 02B0H is FF for VDM, then the VDM address is at 02B1 & 2H. Change CC00 to FC00 for Hoggs Solos. That is the only change needed unless you are using the optional clear screen initialization at address 02A4. If this is 0000C9 then no change is needed, but if there is a call to Sol's clear screen in the "C" block, change it to Hogg's routine - see the source code he furnishes.

Most Wordstar won't use this initialization - that's easy

In answer to Rick Down's letter on using Pencil with Hogg's double density board, he is right the N* version won't work and you would have to rewrite the code. However there is a C/PM version and it works fine since it uses C/PM calls only, actually it is better than the N* version since the bugs have been fixed.

Clyde Steiner

Dear Stan,

Sorry that I haven't written sooner but I've been busy trying to interface my Sol to a Cope 1030 printer. As you can see from the typing in this letter, I still haven't succeeded!

I have been using Processor Tech.'s PILOT since it was introduced about four years ago. I cursed its lack of disc commands but my wife and I plodded along writing innumerable small teaching programs for our son. None of them was worth publishing but each one served a purpose. Not everything worth doing is worth doing well!

Well Ellis Computing, 600 41st Ave., San Francisco, 94121, has come up with a CP/M disc based version called Nevada PILOT. Not only does it come on almost all disc formats, it has a built in initialization program that works for the SOL or a CUTS board with VDM!

Nevada PILOT is revised by none other than John Starkweather, the creator of PILOT. Any readers familiar with Processor Tech.'s version will recognize this one immediately. The manual, in fact, is largely a repeat of the earlier P.T. manual. This, however, is just the starting point.

The most obvious improvement is the disc commands. Both programs and data can be saved and copied from disc. This is most useful when letting a student use the computer for computer assisted instruction. You can easily write the program so that the student's incorrect answers are recorded to let you see where he needs help.

There's a text editor almost identical to that in Processor Tech.'s ECBASIC with full page editing. If you are still struggling along with CP/M's ED, you will find PILOT's editor good enough reason to buy the package.

There are a lot of I/O related tricks such as inverse video and cursor addressing that let you make interesting displays to accompany your educational programs. There are even commands to let you control voice units and video cassette recorders. As I have neither of these and as none of my friends will lend me his, I haven't tested them with Nevada PILOT but the rest of the package is so professional, I'm sure these will work too. Most importantly, all these features are readily available for even someone with no programming experience.

Don't let the fact that PILOT is often used for teaching put you off. There are lots of places where you could use teaching programs outside a school. Write programs to take histories in your medical, dental or law office. Use it to train your office personel in using that new computer you just intorduced. Use it to show your non-computer friends how friendly your computer can be.

As you can see, I am impressed with this program. I think anyone else who tries it will be too.

I'll get back to you later, Stan, when I have my Cope running and Write an article on how to do it.

Sincerely,

Bruce Evans

Bruce Evans

REPRINT

CP/M Plus™ will offer 2-5 times performance of standard CP/M 2.2

With 8-bit hardware sales still running strong, Digital Research's industry standard CP/M 2.2 operating system continues as the 8-bit O.S. leader. More than 700 OEMs are now using CP/M 2.2 and more than 3000 application programs have been written for that operating system by ISVs. According to industry experts, some 700,000 microcomputer users are taking advantage of 8-bit CP/M capabilities at the present time.

And now Digital Research is readying the introduction of its newest version of the favorite 8-bit operating system -- CP/M Plus. John Rowley, DRI chief operating officer, commented, "As the leading OEM supplier of microcomputer software, we feel it is critical to enhance our mature product offerings consistent with new hardware capabilities, allowing our customers to be more competitive with their product offerings. CP/M Plus, version 3.0, gives our customers two to five times the performance of CP/M 2.2."

8-bit market still expanding

Rowley expects the 8-bit market to remain strong through the 1980s, citing as evidence the appearance of National Semiconductor CMOS Z80 equivalent devices, the Zilog Z800 and the Intel 10 mHz 8085, as well as the strong trend toward larger memory in 8-bit hardware.

Why introduce a new version during a period of continuing CP/M 2.2 success? Kathy Strutynski of DRI's Operating Systems Strategic Business Center explained: "We recognize that there is hardware out there now that can do a lot more. Many companies are offering more than 64K, and we want to give them the performance that is possible with more memory."

CP/M Plus will be fully upward compatible with CP/M 2.2 at the functional level; that means that application software will require no changes to execute under CP/M Plus. Two versions of the new O.S. will be available: a smaller non-banked, resident system and a larger bank-switched system providing increased space for application programs.

"We've taken the valuable advice of our ISVs, OEMs and end users, and combined it with our own years of experience with CP/M, to come up with the design of the 3.0 version," Strutynski said. "It's a greatly improved product with better performance and greater user friendliness. Our major design objective was simplicity, and I think we've achieved it."

ISVs and end users alike will quickly note the increased friendliness of CP/M Plus which features automatic disk login allowing a diskette change without a warm boot; date and time stamping for file creation, access and update; a HELP facility, and a console input /output redirection facility (input can be read from a file and output written to a file). The USER facility also is enhanced to allow the operator to store commonly used transient programs under USER 0 and access these programs for any USER number.

Features for ISVs

New functions have been added to the O.S. to allow application programs to:

★ determine the amount of free space on a selected drive;
★ chain to another program;
★ trap O.S. errors;
★ truncate a file; and
★ load and overlay program segments.

Hardware requirements for both the banked and non-banked versions of CP/M Plus, version 3.0, include an 8080/8085 or Z80 microprocessor, at least one disk and a console. The banked version will require a minimum of two banks with the top region (4-32K) in common and the low region bank-switched. The non-banked system requires approximately 4K more than CP/M 2.2.

More information on the new CP/M Plus will be forthcoming from Digital Research in the months to come. Watch for it in *Digital Research News*, upcoming *ISV Forums*, at COMDEX or write for product literature from Digital Research Customer Service.

An overview of CP/M Plus and assistance with its implementation will be offered in a seminar to ISVs Dec. 6 and 7 on the Monterey Peninsula. Although this seminar is filled, Digital Research will repeat this educational offering Feb. 5 and 6. Information is available from Seminar Coordinator Peggy Anderson, 160 Central Ave., Pacific Grove, CA 93950; (408) 649-3896.

Digital Research ISV FORUM, page 5

November 1982

EDITOR'S NOTE: ISV = Independent Software Vendor, that is, someone who writes and markets his own software.

your commands

```
CHECK        CALCULATE UNIQUE IDENTIFING CHECKSUM FOR DISK FILE
COPY         COPY FILES BY MENU NUMBERS
DIR          LIST DIRECTORY IN ANY SELECTED FORMAT
DISK         SHOW DENSITY AND STORAGE PARAMETERS FOR DISKS
DS           DISPLAY MEMORY IN ASCII,HEX,DECIMAL,AND BINARY
                EQUIVALENTS, SINGLE STEPPING IN ANY DIRECTION
                AND ALLOWING CHANGES IN ANY OF THESE FORMS
DUMP         DUMP ASCII EXACTLTY AS CONTAINED IN MEMORY
DUMPA        DUMP BLOCK OF MEMORY IN FORMATED ASCII LISTING
DUMPH        DUMP BLOCK OF MEMORY IN FORMATED HEX LISTING
DUMPX        DUMP BLOCK OF MEMORY IN FORMATED HEX & ASCII
ERA          ERASE FILES BY MENU NUMBERS
EX           JUMP TO ADDRESS ENTERED AND EXECUTE A PROGRAM AND
                THEN RETURN TO POWER
EXIT         EXIT  POWER  AND  JUMP  TO  CUSTOM  ADDRESS
                (CURRENT DEFAULT IS 00 FOR CP/M WARM BOOT)
FILL         FILE BLOCK OF MEMORY WITH HEX BYTE
GO           LOAD AND AUTO EXECUTE A PROGRAM AT ANY MEMORY LOCATION
GROUP        SHOW ACTUAL DISK LOCATIONS OCCUPIED BY ANY CP/M FILE
JP           JUMP  TO ADDRESS ENTERED TO  EXECUTE  PROGRAM  OR
                ROUTINE AND THEN RETURN TO CP/M WARM BOOT
LOAD         LOAD A FILE FROM DISK TO ANY LOCATION IN MEMORY
LOG          LIST THE LOG OF POWER DEFAULT SETTINGS
MOVE         MOVE BLOCK OF MEMORY TO SPECIFIED LOCATION
READ         READ ANY TRACK & SECTOR TO ANY MEMORY LOCATION
READGR       READ ANY CPM FILE GROUP FROM DISK
RECLAIM      RECOVER  PREVIOUSLY DELETED FILES FROM  DISK  AND
                RECLAIM THEIR NAMES IN DISK DIRECTORY
REN          RENAME FILE
RUN          RUN A PROGRAM FROM POWER
SAVE         SAVE A FILE TO DISK FROM ANY LOCATION IN MEMORY
SEARCH       SEARCH MEMORY FOR ASCII,HEX OR BOTH USING WILD CARDS
SETDIR       SET FILE TO $DIR (LISTED IN DIRECTORY)
SETRO        SET FILE TO $R/O (READ ONLY)
SETSYS       SET FILE TO $SYS (NOT LISTED IN DIRECTORY)
SETWR        SET FILE TO $R/W (READ AND WRITE)
SIZE         LIST FILE SIZE IN SECTORS AND KILOBYTES
SPEED        SET DISPLAY SPEED OF LISTINGS ON CONSOLE
STAT         LIST STATISTICS ON REMAINING FREE DISK SPACE FOR
                DRIVES THAT HAVE BEEN ACCESSED (USE ^C FIRST TO
                ASSURE UPDATED STATISTICS IF DISKS HAVE CHANGED)
TEST         TEST DISK MEDIA FOR GLITCHES AND PRINT UNIQUE
                CHECKSUM FOR ENTIRE DISK.COLLECT BAD BLOCKS IN A FILE
TYPE         TYPE A TEXT OR ASCII FILE EXACTLY AS ON THE DISK
TYPEA        TYPE AN ASCII DISK FILE IN FORMATED  AND NUMBERED
                16 CHARACTER LINES
TYPEH        TYPE A .COM OR OTHER BINARY FILE IN FORMATED AND
                NUMBERED 16 HEX CHARACTER LINES
TYPEX        TYPE A .COM OR OTHER BINARY DISK FILE IN FORMATEL
USER         SET FILES TO AN EXCLUSIVE USER AREA (0 TO 31)
USR1            SPACE FOR FOUR USER CONFIGUABLE
USR2            CUSTOM COMMANDS IS PROVIDED.
USR3            ENTER YOUR OWN JUMPS
USR4            TO YOUR OWN ROUTINES.
WRITE        WRITE MEMORY TO ANY TRACK AND SECTOR DISK LOCATION
WRITEGR      WRITE ANY CPM FILE GROUP TO DISK
XUSER        SET NEW USER AREA TO RECIEVE TRANSFERRED FILES.
                THRITY TWO DESTINATION USER AREAS AVAILABLE.
?            LIST AVAILABLE COMMANDS TO SCREEN
```

*CP/M is a trademark of Digital Research

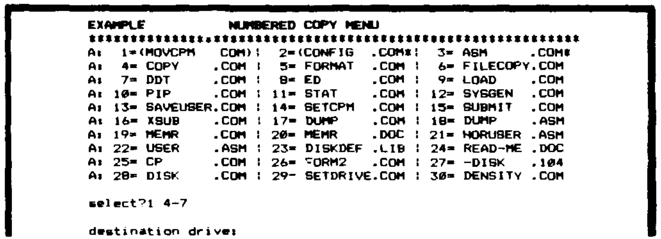
don't type this EVER again

PIP A:=C:AB,A:=B:TEXT

NOW

. . . The first super program that puts YOU in control of CP/M.*

let POWER do it



```
EXAMPLE            NUMBERED COPY MENU
**************************************************************
A:  1=(MOVCPM   COM)!  2=(CONFIG  .COM*!  3= ASM     .COM*
A:  4= COPY     .COM !  5= FORMAT  .COM !  6= FILECOPY.COM
A:  7= DDT      .COM !  8= ED      .COM !  9= LOAD    .COM
A: 10= PIP      .COM ! 11= STAT    .COM ! 12= SYSGEN  .COM
A: 13= SAVEUSER.COM ! 14= SETCPM  .COM ! 15= SUBMIT  .COM
A: 16= XSUB     .COM ! 17= DUMP    .COM ! 18= DUMP    .ASM
A: 19= MEMR     .COM ! 20= MEMR    .DOC ! 21= NORUSER .ASM
A: 22= USER     .ASM ! 23= DISKDEF .LIB ! 24= READ-ME .DOC
A: 25= CP       .COM ! 26= FORM2   .COM ! 27= -DISK   .104
A: 28= DISK     .COM ! 29- SETDRIVE.COM ! 30= DENSITY .COM

select?1 4-7

destination drive:
```

just pick the numbers

You can transfer files without having to type in filenames. You can select files by number from a screen menu, and erase files using that same numbered menu. You can type files to the screen or to the printer by specifying their menu numbers.

Power's menu function protects you from your own typing errors.

ONLY $149

POWER

makes CP/M a joy to use

COMPUTING! 2519 Greenwich,
San Francisco, CA 94123.

save $600

● ● POWER is a computer program that can save the average CP/M* user over $600 a year if they value their time at $6 per hour.

The program not only saves time and money, but it can end frustration. If you have ever accidentally erased a file or mistyped one of a series of names while copying from disk to disk, you'll want to see what ● ● POWER can do for you.

The program is a series of 50 user friendly houskeeping CP/M* programs in one 12k package. They range from a menu function to save you from typing mistakes to the most sophisticated monitor available for a micro computer.

Using the ● ● POWER program means you never again have to type a program or file name in normal computer housekeeping chores. You transfer files without typing file names. The computer does it for you. You then select the files by number from a screen menu. You erase files using that same numbered menu. You type out files to screen or printer by menu number. You can even run programs from the numbered menu.

The purpose of the ● ● POWER menu function is to protect you from your own typing mistakes. Furthermore the program traps that unwelcome BDOS ERROR ON...More importantly, it allows you to correct errors with a reclaim function that restores accidentally erased files. A disk test function salvages glitched or flakey disks by gathering bad sectors into a special invisible file.

The user orientation of the program's design allows you to change disks at will and switch between densities. You can now operate your CP/M* computer without a system disk in drive A.

For the advanced programmer, there are monitor commands to read and write to any selected track or sector from any location in memory. You can fill memory, move memory and single step in any direction, entering Ascii, Hex, Decimal, or Binary. You can search memory for code or letters, using wild cards at will.

If that isn't enough, you can load and execute programs any place in memory, not just at CP/M's* standard 100 address.

The 50 commands are fully explained in the alphabetized 60 page manual.

● ● POWER is hardware independent and will operate in any standard CP/M* system including Apple (Z80 card), Osborne, TRS-80 conversions, S100 North Star, Vector, Morrow, Godbout, Ithaca etc. It is available in all disk formats at $149 from: COMPUTING!, 2519 Greenwich, San Francisco, CA. 94123

1) SOL and S-100 Users Memory Bonanza!!!
Here's your chance to purchase Processor Technology's NKRA Dynamic Memory Board fully assembled, tested and burned in. Can be configured from 16KB to 64KB in 4K increments(for a SOL with ALS8, 0 to 48KB, for SOL without ALS8 0 to 48KB, and 52KB to 64KB). Has invisible refresh and does DMA(board has been tested with GODBOUT DISK 1 DMA Disk Controller). Latest rev.P.C.B. Complete with manual for only $250, plus tax(Cal. residents-6.5%) and shipping. limited supply. Contact Bob Demaret, 586 Brighton Way, Livermore,Ca. 94550, or call (415) 443-9535 after 2PM.

2) NKRA Parts for sale
Delay Line plus all 5 ROMs Programmed.$45 plus tax and shipping.
Manual (reprinted).$10 plus tax and shipping.
All 3 above for $50 plus tax and shipping.
Very limited supply of delay lines.
Contact Bob Demaret, 586 Brighton Way, Livermore, Ca. 94550 or call (415)443-9535 after 2PM.

URGENT--
Looking for SOL Character Generator, part # NITRON NCM 6574
Call anytime (message phone), Gardner Bride (415)892-6149.

FOR SALE

SOL 20 with 24K RAM, 5K BASIC, Extedded Cassette BASIC, no monitor.
Charles H. Beineman
309 Camborne Dr.
Englewood, OH 45322

FOR SALE

(A Bargain !)

PT Sol, Helios II System with 9" Sanyo VM 4092 Monitor and Okidata 110 Printer. Printer is RS 232 with tractor. Has Sol Rev. D with 48 K memory, PTDOS 1.5 (mod 2), serial port driver for printer, Extended Disc basic and several games, etc. Excellent condition and very complete documentation. This is now "surplus" to my needs and will consider any reasonable offer.

H.W. Chamberlain, 1253 Hawthorne Street, Alameda, CA 94501
(415) 521-0221

HELIOS DISK SYSTEMS FOR SALE

Two (2) complete Helios Disk systems for sale at an enormous savings. $1000.00 buys disk drive, boards, full documentation and a ton of software (PTDOS, languages, games, utilities, etc.) Will sell the board sets and drives seperately. $200.00 for the board sets, $875.00 for the drives. All are at latest rev. Drives have been recently overhauled and aligned at the factory with latest revs installed.
Contact Grayson Evans, 946 SW Westwood Dr., Portland Or. 97201
503-244-9172.

I have a SOL 20 Computer with 48 K memory, and assorted software that I would like to sell. I am hoping that you might circulate this to your users group in hopes that if someone is interested they will call me.

The Computer is in good condition and has recently been checked out and repaired by the service department of Computer Mart of New Jersey.

I will sell the Computer, Software, 12" Video Monitor, and Cables for the best offer over $1,000.00 + Freight. I can be reached during business hours at 809-790 5001, or invite inquiries at the address above.

Thank you in advance,

Arthur A. Silver
c/o Pueblo Communications Inc.
Radio Stations WQII/WZNT
G.P.O Box 71398
San Juan, Puerto Rico 00936

Arthur A. Silver

FOR SALE-Helios 2 Slot W/PTDOS, including BASIC software.
Mario Pamatmat, P.O. Box 855, San Francisco State University
Tiburon, CA 94920

FOR SALE - SOL-20, 24k RAM, 5K BASIC, Extended Cassette BASIC, no monitor. Also a "bare bones" computer that could be upgraded.
Charles H. Beineman, 309 Camborne Dr., Englewood, OH 45322
(513)836-6056.

WANTED: SOL P.C. board, preferably in working condition, but I will consider any condition.
Jim Williams, 2346 Cowan Blvd. Apt.304, Fredricksburg, VA 22401
days(703)371-8316, evening (703)373-2969.

FOR SALE: P.T. SOL-20 (Rev B) - complete with 32K RAM, parallel and serial ports, and all manuals. Will also include cassette software if desired. $600.00 or make me an offer.
Royce D. Bacon, 8942 W. Lawrence Ave., Milwaukee, WI 53225,
phone (414)462-3418.

EMERGENCY-Looking for Per Sci 270 disk drive - call Ron at (206)225-8267.

FOR SALE - SOL and Helios
Charles Terry, home (713)666-1316, business(713)666-3515.

FOR SALE - Helios 2 Slot disk with controller board for SOL.
$900.00 Bob Hogg, 25651 Minos St. Mission Viego, CA 92691,
(714)770-2168.

TABLE OF CONTENTS

EDITOR'S COLUMN

by Stan Sokolow

I'm sorry for the lateness and thinness of this issue. I've been hoping that members would send in a few more articles, but it seems that everyone is busy with other things or with (heaven forbid) another computer. So we barely managed to put together an acceptable issue this time. Please take this as a request to go right over to the word processor and knock out a story for us on anything you've done or learned about computers, rumors, new technologies, bargains, programming or fix-it tips, etc.

For example, here's a tip mentioned in a prior issue but worth repeating. You can refresh a faded printer-ribbon quickly and cheaply by opening the plastic case and squirting in some WD-40 lubricant for a couple of seconds. Allow the ribbon to sit for a day or so to let the oil diffuse evenly throughout the fabric. The light oil mobilizes the ink from the unused parts of the ribbon and also keeps the matrix printing wires lubricated. Got any other tidbits like this? Send it in.

As time goes by, the function of Proteus seems to be slowly evolving and once again I am re-evaluating. Initially, we were all starved for information about our "beloved Sol" (as Stan Veit calls his). But by now, there is such a bombardment of information about microcomputers and such an overwhelming onrush of new machines, that it seems we are left in the dust if we stay with the Sol.

Of course, the Sol is still as functional today as it ever was, and the utility of your Sol to you will depend upon what you want it for. As a "classic" computer, I'm sure that it will someday be a valuable collector's item. In fact, I wish I had the money to invest in buying up other people's "junk" computers, like the original Altair, Imsai, Sphere, SWTPC, etc., to simply store them away for thirty years. Just think of the value of Harrah's antique car collection and translate that into an antique computer collection.

When the Z-80 came out, everyone with an 8080 got worried that their machine was obsolete. I had to restrain myself, too. It is easy to get caught up in the mania of the "latest and greatest" syndrome. More recently, it was the "8088: 16-bit chip on the 8-bit bus" craze, especially with IBM entering the personal computer market with an 8088 computer. But, benchmarks have shown the 8088 is not so swift with the software out there now. My Sol/Helios will still run circles around lots of new machines with those slow floppy disks, in spite of having only a 2-megaHertz 8080 microprocessor.

Well, enough rambling. On with the show. I'll have more to say in the next issue, which I promise will be out before the end of this year.

PROTEUS/NEWS

A news journal for owners and users of Processor Technology Corporation computer equipment. Published by Proteus, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483,USA, telephone (415) 368-2300.

Submit items for publication to Proteus News, Attn: Stan Sokolow, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483, USA. Please make submissions as camera-ready as possible by using a fresh, black ribbon and typing single-spaced.

FROM:
PROTEUS
1690 WOODSIDE ROAD, SUITE 219
REDWOOD CITY, CALIFORNIA 94061-3483
USA

Stephen Maguire

2430 N. Dodge Blvd.#250
Tuscon, AZ
85716

PROTEUS / NEWS

AN INDEPENDENT NEWSLETTER FOR OWNERS AND USERS OF PROCESSOR TECHNOLOGY CORPORATION COMPUTERS

FORMERLY SOLUS NEWS

1982 - FOURTH QUARTER PUBLISHED BIMONTHLY BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA

VOL. 5 # 4
SINGLE ISSUE ISSUE....$7.50 (US)
SINGLE ISSUE ISSUE....$9.50 (FOREIGN)

UCSD p-SYSTEM VERSION IV.1 UPGRADE

Earlier this year, I nearly fell over when I received a package from Softech Microsystems containing an entirely new set of p-System disks and a new manual. I thought it had to be a mistake and that I would soon see a bill.

But, when I looked inside, I found an invoice for zero dollars and a note that says,

> "Enclosed, please find your IV.1 Upgrade.
> "All end users who purchased a system or add-on product after December 14, 1981, and before the release of IV.1, October 18, 1982, receive a version IV.1 upgrade for no additional charge.
> "Version IV.1 includes new features and enhancements as well as general maintenance fixes."

Now, how about that! All that I ever received from Digital Research was a few pages of paper telling me how to incorporate a handful of CP/M corrections myself. Here's how to treat the purchaser right. Thanks, Softech Microsystems.

Some of the new features are as follows:

The symbolic debugger allows setting breakpoints by source program line number instead of p-code instruction location. Also, variables can be examined by name rather than by location.

Subsidiary volumes can be created on disks. This allows a disk to be divided into many areas, each acting like its own disk with directory and files. The system is smart enough to know that when the removable media in the disk drive contains more than one volume, all of the subsidiary volumes go along with the media when it is removed or replaced. Each volume can contain up to 16 megabytes, with the potential total online storage being more than 1700 megabytes. [That ought to take care of you for a while.]

Error messages from the system can be designated to appear on any specified line on the console. You can alter the error messages, so they can be more meaningful to the user, such as when the user doesn't speak English. (Does CP/M come in foreign language versions?)

The USES feature of the compiler, which lets you use portions of library routines, has been made more efficient in memory space.

8-bit character codes can be handled from the keyboard, rather than just 7-bit ASCII.

The system prompt line can be replaced by any action or menu you wish. Whenever the prompt would be displayed, the system executes a program called SYSTEM.MENU, which you may provide. This is useful for creating a turnkey system where users are totally unaware of the underlying p-System. (In other words, absolutely no jargon will appear on the screen to confuse the end-user.)

A performance monitor allows the user to track such automatic system activities as removal of code segements from

CON'T ON PAGE 2

US POSTAL SERVICE OFFERS ELECTRONIC ORIGINATED MAIL

E-COM (Electronic Computer Originated Mail) is a new, little known service of the United States Postal Service for the convenience of volume mailers who generate first class mail from data in a computer. It is intended for users who mail at least 200 letters at a time, however, smaller amounts will be accepted and billed for 200 piece minimum charge.

With E-COM service, the user transmits the data for the letters from his or her computer directly to the postal service's computer, where it is printed, enveloped, and introduced into the regular first class mail system for delivery. Service began on January 4, 1982, and it is now available to all certified users. (Certification requires verifying that your hardware and software can transmit acceptable messages. This is done by having you send a sample message. If no errors are processed, certification is granted.)

Businesses who send statements periodically are ideal candidates for E-COM, but it appears than anyone can use the service, with credit approval and appropriate equipment. The price is reasonable: 26 cents for a one-page letter, 31 cents for a two page. This includes printing, paper, envelope, and postage. There is also an annual fee of $50 to establish and maintain an account with the post office.

Messages can be one of three types: Single address messages (SAM), Common text messages (COT), and text insertion messages (TIM).

SAM's include the text and destination address for each letter. COT's contain one block of text to be sent to each of many addresses. TIM's allow plugging-in of information into a skeletal letter, customizing it for each address. The addresses and unique information are provided by the sender along with the skeletal form.

Text may contain any mix of lines and alphanumeric characters, using a symbolic code explained in the E-COM manual. Letters are printed on white bond paper, 8-1/2" by 11", without perforated edges or sprocket holes. The envelope will be delivered in an envelope with a prominent "E-COM" legend.

Transmission of data requires a 300, 1200, 2400, or 4800 baud modem. The protocols are described in the E-COM manual. Printing is dot-matrix type.

E-COM is available from 25 serving post offices around the USA. That is, you must call a phone number in one of these service areas to transmit data to the computer. The mail can be addressed to anyone in the US. Since your return address appears on the letter, I assume it will be shown through a double window envelope. Unless regulations have changed recently, only single window envelopes are permitted in international mail, so I assume this is only available for domestic addresses.

If you have business mailings, or if you are in a club with over 200 members, or if your organization sends out political action notices, think how much easier this may be without the hassles of printing, envelope stuffing, licking and sticking. Since postage alone is 20 cents, the postal service is only charging you 6 cents for the printing, paper, envelope,

CON'T ON PAGE 2

CON'T FROM PAGE 1 -(UCSD p-SYSTEM VERSION IV.1 UPGRADE)

memory, procedure starts and stops, etc. This will let the sophisticated application programmer fine tune his system and monitor time-consuming sections of code.

File names can contain a wild-card code that matches a range of names in a very flexible way; much more sophisticated than CP/M's ambiguous file naming facility or PTDOS's convention.

Units (library programs) are provided for accessing file directory information and system information in a machine independent fashion.

UCSD p-SYSTEM FOR HELIOS

by Stan Sokolow

From the user's point of view, portability to new machines is a very important consideration in buying programs, in my opinion. The longer you use your system, the more obsolete it becomes. As hardware becomes cheaper to make, the cost of upgrading your computer hardware to an entirely new machine becomes more reasonable. But what will far overshadow this, is the cost of re-establishing your software base.

Sure, CP/M is so widespread that it is practically a universal operating system. But can you take your CP/M programs written for you 8080 machine and run them on a 68000 microprocessor computer with CP/M-68K? No. And sure, 8080 code can be translated into 8086 code by programs for that purpose, but only if you have the source code. So, if you build a library of programs you've purchased, like editors, word processors, database managers, etc., can you hook up your old and new computers, send the programs over to the new machine and be all set to run them? No, of course not. So what do you do? Buy new software.

Now, who cares about this problem? Not the computer manufacturers, because they prefer to lock you into their product line anyway. Not the independent software houses, because they would be delighted to re-sell the same package to you for you new machine's microprocessor. And not the users, because they don't know the difference and have a short-range view of their needs.

But you, as an experienced user, should realize this by now. Just look at what Apple has introduced in their Lisa computer. Here's a big leap forward in ease of use, software integration, fail-safety, etc. And what does IBM have in the works? Not to mention the Oriental tidal wave of machines that is bound to follow.

Just imagine how crazy the audio world would be if you had to buy a new record collection every time you upgraded your audio system to a new model. Well, that's where we are in computers.

Now, it may happen that every machine will provide an 8080/8085/Z-80 compatible processor along with the host microprocessor, but I doubt it. Some do now, and certainly this will continue for a while. But I don't want to count on it.

The best way I can protect myself against this problem, as I see it, is to use software that is portable to other dissimilar computers. And right now, the only computer operating system that meets this requirement is the UCSD p-System.

As I explained in a previous issue, the UCSD p-System is an operating and programming system that is written in a pseudo-machine language. This language is not actually accepted by any microprocessor, although some have been microprogrammed to act like they run that object code. But each computer that uses the p-System contains an interpreter program that reads the p-code instructions and simulates the pseudo-machine using subroutines in the interpreter.

This runs slower, but when the machines are running faster and faster each year, what difference does it make? It certainly runs faster than BASIC interpreters. Moreover, Softech has native-code generator programs which translate portions of your p-code into faster running native machine language instructions, so you can speed up critical areas of code you write. Most programs are waiting for input/output much of the time anyway.

The p-System has had a slow, difficult life in the marketplace, but it has achieved a high degree of acceptance among major manufacturers, which I think will keep it viable. As the large corporate users move into local networking more and more, the universality of the p-System may give it a competitive advantage: Any computer in the network will be able to run the very same object code program as any other, since they all will accept the p-code regardless of microprocessor. One network will be able to mix Z-80's, 8086's, 68000's, etc., and have only one program library for all of them to use. Even the representation of real numbers, which varies among machines, has a cannonical form in the p-System when it is written on disk.

It seems to me that networking is a natural for the p-System because Softech has made portability the number one concern. It is available for the following microprocessors: 68000, 8086/8088, Z80/8080, PDP 11 (various models), LSI-11, 6502, and 9900. This just about covers all of the current machines, even the new Apple Lisa which uses a 68000.

So, although I love PTDOS on my Helios, and although CP/M is ubiquitous and cheap and I have it already, I have invested in a p-System and have arranged to have it adapted to my Sol/Helios computer. (I wish I had time to do it myself.) Drew Rogge, who worked on the UCSD p-System implementation at Processor Tech before they went under, has looked over the documentation and has agreed to work on installing the latest version (IV.1) on my Sol.

After he and I have gotten it to be really nifty on the Sol/Helios, such as recognizing the enhancements we have for the Sol (the usual Sol drivers, 24x80 screen, keyboard customizer, 64K memory space, etc) I will let you know through the newletter. If there is enough interest, I will buy the license to distribute the p-System to other Proteus members.

The system is expensive when purchased directly from Softech, although quite comparable to the cost of similar software for a CP/M environment. But there is enough markup in the dealer cost that Proteus can afford to give members a good discount below retail.

I have long-range plans for my computers, and I don't want to find myself painted into a corner every 5 years. Although I wasn't happy with earlier versions of the p-System, I feel that version IV is an excellent system and definitely here to stay. That's why I'm not enthused about working on transporting PTDOS to other disk drives. I don't think it makes sense in the long run. And in my mind, neither does it make sense to put a lot of money into purchased CP/M software.

If you have any views on this, I would like to hear from you for publication in the next newsletter.

CON'T FROM PAGE 1 (US POSTAL SERVICE OFFERS ELECTRONIC ORIGINATED MAIL)

and handling. Not bad, I think.

For more information, write Director of E-COM operations, United States Postal Service, 475 L'Enfant Plaza, SW, Washington, DC 20260-7140.

ELECTRONIC SPREAD SHEET PROGRAM

by Franz Hirner

631 Matsonia Drive
Foster City, California 94404
December 20, 1982

Stan Sokolow
1690 Woodside Road., Suite 219
Redwood City, California 94061

Dear Stan,

Enclosed is a description of a Electronic Spread Sheet program I have been running on my Sol-North Star for some time. I thought that there may be some Sol users who would like a spreadsheet program but can't afford the $300.00 to $600.00 price for some of the commercial programs. Feel free to publish any or all of the enclosed material.

The program listing is in compacted form to allow larger data arrays to be in memory. Since all data is in memory, and the disk is used only for storage, the program could be adapted to run under PTC Basic of other basics without much difficulity.

For those who don't want to manually enter the program, I would be happy to supply copies on North Star Disk along with a copy of the Description for $ 25.00. I would need to know if a member wanted single or double density. I could also provide a uncompacted listing for those wanting to convert the program to PTC Basic.

I recieved several requests for the General Ledger Program submitted earlier which I was happy to provide. I have a Data Base Program in operation, but haven't had the time to write a description yet. When I do, I'll be glad to make it available to the group.

Do any of the other members have any application programs they would make available? It dosen't make much sense to reinvent the wheel.

I do have a Integral Data Model 440 Printer I'm not using and would like to sell preferably in the Northern California to avoid packing and shipping. The price is $500.00

Does anyone have a copy of North Star Pascal for the double density disks they would like to trade for a single density version. I'm unable to figure out a way to convert my single density Pascal to operate on the double density drives.

Thank you.

Very truly yours,

Franz J. Hirner

Franz Hirner

Program Description:

The Electronic Spread Sheet is a program allowing the user to create his or her own mathimatical model of almost anything that can be represented by numbers on a columer spread sheet.

The size of the the spreadsheet can be varied to suit the user depending on the application at hand. The variables R and C appear near the front of the program. These variables control the number if rows and columns of the spreadsheet. The user must allow at least 10 columns for the width of the spreadsheet. By changing the value of these two variables, the user may configure the size of the spread sheet to suit his requirements. When implemented on a Sol-20 with North Star DOS located at D000 Hex, and Basic located at 0 Hex, and the 48K memory, there is room for about 1000 data elements when the program is in its compacted from, that is all spaces and Remards removed from the program text.

There is also a print format statement adjacent to the R and C variables which controls the number of decimal places in numerical answers. This may also be changed to suit the needs of the user but the total length allocated to each number should be left at twelve spaces, otherwise the format of the output will not line up properly with rows and columns.

There are three principal screen displays in the ESS program.

1. Main Menu
2. Data Display
3. Parameter Display

MAIN MENU

The main menu provides a directry of the following functions:

1. Retrieve Data File
2. Save Data File
3. List Disk (not implemented at this time)
4. View and/or Modify Parameters
5. View and/or Modify Numerical Values
6. End

Items one, two, and three are self explanatory.

Item 4, Parameters is the only way of entering calculation parameters or other codes defining how a data element is obtained. Parameter definitions allow data to be input, calculted from other data elements, etc. Also, print suppression codes can also be defined for each Parameter to eleminate unwanted output on either the monitor screen or hard copy printout.

For purposes of discussion, the user must understand that reference to a data element must be identified by the exact row and column. Data values are entered from the Data Dcreen and Parameter Values are entered from the Parameter Screen. The parameter codes must be placed in the position of the array where the user wants the numerical result of a given function to appear.

CON'T ON PAGE 4

CON'T FROM PAGE 3

A Parameter has a total possibility of seven components. Not all of the components are required or used for each parameter. The parameters are as follows:

1. Parameter Code - two letters that define the code.
2. R1 - Reference pointer to Row number one.
3. C1 - Reference pointer to Column number one.
4. R2 - Reference pointer to Row number two.
5. C2 - Reference pointer to Row number two.
6. V1 - Variable number one.
7. V2 - Variable number two.

All of the parameters are available for every element of the array, although all are not always used. When the user enters a code, the ESS will automatically prompt for only the necessary data required for the function being entered by the user.

All command codes are one character codes entered by the user from the keyboard when either the Data or Parameter Screen is displayed. A command code is necessary to enter any data into the program. A (CR) is not required after the command key, but is necessary after each data element is entered. Following is a listing of the command codes. A "D" or "P" follows the code to indicate which screen from which screen the code is operable.

Command: 1 (D)

Enters a vertical scroll rate. Initialized to one at program start.

Command: 2(D)

Enters a horizontal scroll rate. Initialized to one at program start.

Command: (3) (D)

Allows for the insertion of a new row on a existing spreadsheet. All numerical data and parameter data are automatically shifted and recalculated. This saves much time when revising an existing spread sheet. Although the Parameter definitions are changed and data elemnt references are revised, the user should use care to check the Parameters whrn the insertion of a new line results in a change in the structure of the program.

Command: (4) (D)

Allows the deletion of a row from the spreadsheet. Data and Parameters are revised as mentioned above.

Command: (+) (D)

Allows the user to add a numerical value to a data element.

Command: (-) (D)

Allows the user to subtract a numerical value from a data element.

Command: (/) (D)

Allows the user to divide the number in a data element by a numerical value input from the screen.

Command: (*) (D)

Allows the user to multiply the number in a data element by a numerical value input from the screen.

Command: (%) (D)

Allows the user to increase the value in a data element by a numerical value entered from the keyboard.

Command: (Left Arrow) (D,P)

Shifts the screen display to the left by the number of columns defined by the scroll rate.

Command: (Right Arrow) (D,P)

Shifts the screen display to the right by the number of columns defined by the scroll rate.

Command: (Up Arrow) (D,P)

Shifts the screen display up by the number of rows defined by the scroll rate.

Command: (Down Arrow) (D,P)

Shifts the screen display down by the number of rows defined by the scrool rate.

Command: (A) (D)

Analizes the spread sheet and calculates those numeric values which have neen defined by Parameter as a calculated value derived from other elements in the data array. The program examins every data element on the spread sheet precedeing down each row starting with the rows in column 1, followed by the rows in column 2, 3, 4, etc. The user should be sure that when he defines a data element as calculated from other data elements, that those data elements will in fact contain the expected values at the time of calculation. Small spreadsheets with few data elements and few calculations will take just a few seconds to analize. Larger spreadsheets with mny calculations will take more time, depending on the size of he spreadsheet and the number of calculations to be performed.

Command: (C) (D)

Allows the user to enter the Titles for each column on the spreadsheet.

Command: (D) (D,P)

Allows the user to select the area of the spread sheet displayed on the screen. The user will be prompted for both the row and column to appear at the upper left hand corner of the screen.

Command: (E) (D,P)

Allows the user to enter either numerical data of parameter data depending on the current display screen. The user will be prompted for the row, column, and input value. If in the

CON'T ON PAGE 5

CON'T FROM PAGE 4

Parameter Display, the user will be prompted also for the Parameter definition (AA,AD,SU, etc), and the necessary references required by the parameter entered, but first the existine Parameters will be displayed. When entering Parameters, the user will be prompted for a second column number. If a number other than the identical value entered for the first column is entered, the parameter will be entered for the row across all columns referenced and the column reference numbers will be updated automatically by the prooram.

Command: (G) (D)

Provides the user the capability of graphing data either in the monitor screen, or on a hard copy device. Hard copy device codes are based on the Integral Data 480 printer and may have to be modified for other printers.

One of the Parameters is INPUT. This defines a data element as input from the user. Command (I) examins each data element and prompts the user for the numerical input for that data element. The user also has the option of manually using Command (E) to enter numerical data but in some cases Command (I) is more convenient.

Command: (M) (D)

Command (M) allows the user to input values into an array of daa elements regardless of the parameters for those elements.

Command: (P) (D)

Allows the output of the spreadsheet to a hard copy device. The user should keep in mind that the printout will be ten (10) columns wide ending with the right most column on the screen. In some cases it will be necessary to print more that one hard copy page if the entire spreadsheet is to be printed on hard copy.

Command: (R) (D)

Allows the User to enter Row Titles (similiar to C for Columns).
Command: (   ) (D,P)

Command: (S) (D)

Shifts the columns specified by the user one data element to the left. This feature is useful for updating spreadsheets which use the columns for time periods such as days, months, years, etc. Only the Column Titles, and data element values are shifted. The Parameter definitions are unchanged.

Command: (T) (D)

Allows the user to enter a Title for the spreadsheet.

PARAMETER CODES

Parameter Codes are used to define which data elements on the spreadsheet are calculated and how they are to be calculated. It is not necessary for all data elements to be calculated but at the same time, virtually all of the data elements could be defined as calculated. Remember that all calculations are performed in the following order: Column 1, Row 1 to n, Column 2, Row 1 to n, etc. etc. The user must be sure that the data elements defined as calculated will in fact perform the calculations in the expected order. This is not difficult if the user will keep the calculation order in mind.

Following are definitions for the Parameters contained in the program:

AA

Add Array sums the block of Data points beginning with R1-C1 and ending with R2-C2. The result is placed in the home array element.

AB

Absolute stores the absolute value of the data element referenced by R1-C1 in the home data element.

AC

The arccosine of the data element returned in degrees.

AD
Add Data sums two specefic data elements.

AK

Add Constant Adds the constant contained in Varaible1 to the data element referenced by R1-C1. The result is stored in the home data element.

AS

The arcsin of the data element specified returned in degrees.

AT

The arctangent of the data element in degrees.

AV

Average will take the sum of the the data elements referenced by R1-C1 to R2-C2, divide by the number of data elements referenced to obtain the numerical average, and store the average in the home data element.

BL

Blank will cause the position of the home data element to be filled with blanks during output rather than the numerical value which occupies the home data element. Blanks are useful for separating areas of output to provide organization to the ESS and enhance readability. Note that a number may still be stored in this data element even though it will not be printed. This feature may be useful for storing constants in the data array to be used by formulas in calculating other data elements. Only

CON'T ON PAGE 6

CON'T FROM PAGE 5

numbers whose significance to the printed output is minimal should be stored in this fashion.

CO

The cosine of ther data element expressed in degrees.

DD

Divide Data divides the Data Element referenced by R1-C1 by the data element referenced by R2-C2.

DK

Divide by constant divides the data element referenced by R1-C1 by the data contained in Variable1. The result is is stored in the home data element.

GA

Grow Average calculates the average change in value for all of the data elements referenced by R1-C1 to R2-C2. The user must be sure that either R1 and R2 are identical or that C1 and C2 are indentical. The average change in value is then multiplied by the last, repeat last, value referenced and that result is stored in the home data element.

GE

Get Data gets the data presently in the data element referenced by R1-C1 and stores the same data in the home data element.

G1

Get Inverse is similiar to the GE function, except that the sign of the data is changed. Positive values become negative and negative values become positive.

IN

Interest calculates the dollar amount of interest on a loan balance. The principal amount of the loan must appear as one of the data elements of the ESS. The principal amount is referenced by R1-C1. The payment amount must also appear as one of the data elements on the ESS spreadsheet and is referenced by R2-C2. The payment should be expressed for the same period as the ESS column. The interest rate is contained in the Variable1 and should be entered as a whole number, ie. .125 = 12.5% interest. Variable #2 must contain the number of payments made per period. If the column on the spreadsheet represents one year time, but payments are made monthly than the total amount of payments for the year is entered, and the yearly interest rate expressed as a percent is entered. The variable 12 is entered for the number of payments per period. Before performing the calculations, ESS will first calculate the amount of each periodic payment and the periodic interest rate. The dollar amount of the interest is stored in the home data element.

IP

Input merely is a control for the calculation portion of the program. This code, as well as BL or a blank " " signals that no calculations are necessary. The IP code also serves to alert the user that the home data element is a input value rather than calculated.

MD

Multiply Data multiplies the Data Element R1-C1 by R2-C2. The result is stored in the home data element.

MK

Multiply Constant multiplies the constant contained in Varaible1 to the data element referenced by R1-C1. The result is stored in the home data element.

PC

Percent Change is similiar to PE in function, except that the numerical value 100 is subtracted from the result before storing in the home data element. This causes the division value .80 to be represented as -20%.

PE

Percent calculates the percent of one number to another. The data element referenced by R1-C1 is divided by the data element referenced by R2-C2 and the result is multiplied by 100 to convert to a percentage ie. 50%. The result is stored in the home data element.

SK

Subtract Constant subtracts the constant contained in Varaible1 to the data element referenced by R1-C1. The result is stored in the home data element.

SN

The sine of the data element expressed in degrees.

SQ

The Square root of the Data Element.

SU

Subtract Data Element Referenced by R2-C2 from R1-C1. The result is stored in the home data element.

XY

X to the power of Y.

YX

The inverse of X to the power of Y.

ZE

Zero forces the numerical value "0" into the home data element.

CON'T ON PAGE 7

CON'T FROM PAGE 6

```
10V8=1\V9=1\R1=1\C1=1\GOSUB50\LINE100\LINE#1,133\LINE#2,133
20INPUT"Enter Number of Rows ",R\PRINT\INPUT"Enter Number of Columns ",C
30PRINT%#12F2\DIMA$(12)\DIMR1$(12)\DIMC1$(12)\DIMD(R,C)\DIMC$(C*12)\DIMP$(12)
40DIMR$(R*12)\DIMD$(R*C*12)\DIMT$(64)\DIMT1$(64)\DIMF$(12)\DIMV$(12)\GOTO360
50PRINTCHR$(11)\PRINTTAB(14),"* * * ELECTRONIC SPREAD SHEET * * *"\PRINT\RETURN
60P$(3,3)=CHR$(V1)\!V1\P$(4,4)=CHR$(T1)\P$(5,5)=CHR$(V2)\P$(6,6)=CHR$(T2)
70V3=V3*100\P$(7,7)=CHR$(INT(V3/(255*255)))
80V3=V3-(INT(V3/(255*255))*255*255)\P$(8,8)=CHR$(INT(V3/255))
90V3=V3-INT(V3/255)*255\P$(9,9)=CHR$(INT(V3))\V4=V4*100
100P$(10,10)=CHR$(INT(V4/(255*255)))
110V4=V4-(INT(V4/(255*255))*255*255)\P$(11,11)=CHR$(INT(V4/255))
120V4=V4-INT(V4/255)*255\P$(12,12)=CHR$(INT(V4))
130D$((X-1)*C*12+(Y-1)*12+1,(X-1)*C*12+(Y-1)*12+12)=P$\RETURN
140GOSUB200
150A$=P$(3,3)\V1=ASC(A$)\A$=P$(4,4)\T1=ASC(A$)\A$=P$(5,5)\V2=ASC(A$)
160A$=P$(6,6)\T2=ASC(A$)\A$=P$(7,7)\V3=ASC(A$)*255*255\A$=P$(8,8)
170V3=V3+ASC(A$)*255\A$=P$(9,9)\V3=V3+ASC(A$)\V3=V3/100
180A$=P$(10,10)\V4=ASC(A$)*255*255\A$=P$(11,11)\V4=V4+ASC(A$)*255
190A$=P$(12,12)\V4=V4+ASC(A$)\V4=V4/100\P$=P$(1,2)\RETURN
200P$=D$((X-1)*C*12+(Y-1)*12+1,(X-1)*C*12+(Y-1)*12+12)\RETURN
210INPUT"Enter Name of Data File  ",F1$\OPEN#1,F1$\READ#1,T$,R$,C$
220FORX=1TOR*C\READ#1,A$\D$((X-1)*12+1,X*12)=A$\NEXTX\FORX=1TOR\FORY=1TOC
230READ#1,D(X,Y)\NEXT\NEXT\CLOSE#1\RETURN
240INPUT"Enter Name of Data File  ",F1$\A=FILE(F1$)\IFA=3THEN260
250L=(LEN(T$)+LEN(R$)+LEN(C$)+(R*C*15)+R*C*5)/256+2\CREATEF1$,L\PRINTL,L/256
260OPEN#1,F1$\WRITE#1,T$,R$,C$\FORX=1TOR*C\A$=D$((X-1)*12+1,X*12)\WRITE#1,A$
270NEXTX\FORX=1TOR\FORY=1TOC\WRITE#1,D(X,Y)\NEXT\NEXT\CLOSE#1\RETURN
280PRINTCHR$(13)+CHR$(13)+CHR$(14),\IFR1>R-9THENR1=R-9\IFR1<1THENR1=1
290IFC1>C-3THENC1=C-3\IFC1<1THENC1=1\PRINTT$\FORX=0TO3\PRINTTAB(12),%12I,C1+X,\N
EXT\PRINT
300PRINTTAB(15),C$(12*(C1-1)+1,12*(C1+3))\FORX=R1TOR1+9\PRINT%3I,X,
310R1$=R$(12*(X-1)+1,X*12)
320PRINTR1$,\FORY=C1TOC1+3\GOSUB200\C1$=C$(12*(Y-1)+1,Y*12)\IFR1$=F$THEN340
330IFC1$=F$THEN340\IFP$(1,2)="BL"THEN340\PRINTD(X,Y),\GOTO350
340PRINTF$,
350NEXTY\PRINT\NEXTX\PRINT%\RETURN
360GOSUB50\PRINTT$\PRINT"ROWS:",R,"   COLUMNS:",C,"  ",\PRINT"FREE ",FREE(0)\PRI
NT
370PRINT" 1. Retrieve Data File"\PRINT" 2. Save Data File"
380PRINT" 3. List Disk"\PRINT" 4. View and/or Modify Parameters"
390PRINT" 5. View and/or Modify Numerical Values"\PRINT" 6. End "\PRINT
400PRINT"     Enter your choice......",\INPUTA$\IFA$="1"THEN430
410IFA$="2"THEN440\IFA$="3"THENA$="9"\IFA$="4"THEN450\IFA$="5"THEN460
420IFA$="6"THEN470\GOTO360
430GOSUB210\GOTO360
440GOSUB240\GOTO360\GOSUB320\GOTO360
450GOSUB1920\GOTO360
460GOSUB530\GOTO360
470GOTO1900\GOTO360
480INPUT1"  Title? ",T$\FORX=1TO8\T$=T$+F$\NEXT\RETURN
490INPUT1"   Row ? ",A\IFA>RTHENRETURN\IFA<=0THENRETURN\INPUT1"  Title? ",R1$\IF
R1$=""THENRETURN
500R1$=R1$+F$\R$(12*(A-1)+1,12*A)=R1$\RETURN
510INPUT1"  Column No.? ",A\IFA>CTHENRETURN\IFA=0THENRETURN\INPUT1"   Title? ",C
1$
520IFC1$=""THENRETURN\C1$=C1$+F$\C$(12*(A-1)+1,12*A)=C1$\RETURN\R1=1\C1=1
530R1=R1\C1=C1
540GOSUB280\A$=INCHAR$(252)\A=ASC(A$)\IFA$<>"1"THEN560
550INPUT"Enter Horizontal Scroll Value",V9\GOTO540
560IFA$<>"2"THEN570\INPUT"Enter Vertical Scroll Value? ",V8\GOTO540
570IFA$<>"3"THEN640\INPUT"Enter Number of Row to Insert",A3
580R1$="            "
590R$=R$(1,(A3-1)*12)+R1$+R$((A3-1)*12+1,(R-1)*12)
600FORX=RTOA3+1STEP-1\PRINT"ROW =",X\FORY=1TOC\D(X,Y)=D(X-1,Y)\NEXTY\NEXTX\FORY=
1TOC
610D(A3,Y)=0\NEXTY\FORX1=RTOA3+1STEP-1\FORY=1TOC\X=X1-1\GOSUB140\P$=P$+F$
620IFV1>=A3-1THENV1=V1+1\IFV2>=A3-1THENV2=V2+1\X=X1\GOSUB60\NEXTY\NEXTX1\FORY=1T
OC
630P$="  "+F$\V1=0\V2=0\V3=0\V4=0\T1=0\T2=0\X=A3\GOSUB60\NEXTY
640IFA$<>"4"THEN650\GOTO2320
650IFA$<>"V"THEN690\PRINT"VARIABLE ",\INPUT1" R1 ",R1\INPUT1" C1 ",C1
```

```
660INPUT1" R2 ",R2\INPUT1" C2 ",C2\INPUT1" MIN. VAL. ",M1
670INPUT" MAX. VAL. ",M2\FORX=R1TOR2\FORY=C1TOC2
680D(X,Y)=(RND(0)*(M2-M1))+M1\NEXTY\NEXTX\GOTO540
690IFA$<>"K"THEN720\PRINT"CONSTANT  ",\INPUT1" R1 ",R1\INPUT1" C1 ",C1
700INPUT1" R2 ",R2\INPUT1" C2 ",C2\INPUT"  VALUE",A\FORX=R1TOR2\FORY=C1TOC2
710D(X,Y)=A\NEXTY\NEXTX\GOTO540
720IFA$<>"+"THEN750\INPUT1"ADD R1 ",R1\INPUT1" C1 ",C1\INPUT1"ADD R2 ",R2
730INPUT1" C2 ",C2\INPUT"  PLUS VALUE ",A\FORX=R1TOR2\FORY=C1TOC2
740D(X,Y)=D(X,Y)+A\NEXTY\NEXTX\GOTO540
750IFA$<>"-"THEN780\INPUT1"MINUS R1 ",R1\INPUT1" C1 ",C1\INPUT1"MINUS R2 ",R2
760INPUT1" C2 ",C2\INPUT"  MINUS VALUE ",A\FORX=R1TOR2\FORY=C1TOC2
770D(X,Y)=D(X,Y)-A\NEXTY\NEXTX\GOTO540
780IFA$<>"*"THEN810\INPUT1"MULTIPLY R1 ",R1\INPUT1" C1 ",C1\INPUT1" R2 ",R2
790INPUT1" C2 ",C2\INPUT" BY ",A\FORX=R1TOR2\FORY=C1TOC2\D(X,Y)=D(X,Y)*A\NEXTY
800NEXTX\GOTO540
810IFA$<>"/"THEN840\INPUT1"DIVIDE R1 ",R1\INPUT1" C1 ",C1\INPUT1" R2 ",R2
820INPUT1" C2 ",C2\INPUT" BY ",A\FORX=R1TOR2\FORY=C1TOC2\D(X,Y)=D(X,Y)/A\NEXTY
830NEXTX\GOTO540
840IFA$<>"%"THEN870\INPUT1"INCREASE R1 ",R1\INPUT1" C1 ",C1\INPUT1" R2 ",R2
850INPUT1" C2 ",C2\INPUT" BY % ",A\FORX=R1TOR2\FORY=C1TOC2
860D(X,Y)=D(X,Y)*(A/100+1)\NEXTY\NEXTX\GOTO540
870IFA$<>"T"THEN880\GOSUB480\GOTO540
880IFA$<>"C"THEN890\GOSUB510\GOTO540
890IFA$<>"R"THEN900\GOSUB490\GOTO540
900IFA<>17THEN910\C1=C1-V9\GOTO540
910IFA<>23THEN920\R1=R1-V8\GOTO540
920IFA<>1THEN930\C1=C1+V9\GOTO540
930IFA<>26THEN940\R1=R1+V8\GOTO540
940IFA$<>"Z"THEN950\FORX1=1TOR\FORY1=1TOC\D(X1,Y1)=0\NEXTY1\NEXTX1\GOTO540
950IFA$<>"E"THEN980\PRINT"Enter Data: ",\INPUT1"Row?  ",R2
960INPUT1"   Column?  ",C2\INPUT"   Data Value?  ",A\D(R2,C2)=A\GOTO540
970IFP$(1,2)="BL"THEN1080
980IFA$<>"D"THEN1000\PRINT"DISPLAY: ",\INPUT1"Row ",R1\INPUT1"   Column ",C1
990PRINTCHR$(11),\GOTO540
1000IFA<>0THEN1010\RETURN
1010IFA$<>"P"THEN1100\GOSUB50\INPUT"Enter Printer Number",P\IFP<>1THEN1030
1020PRINT#P,CHR$(29),
1030PRINT#P,T$\IFC1<>1THENPRINT#P\IFP<>1THEN1040\PRINT#P,CHR$(31),
1040PRINT#P\IFC1>C-9THENC1=C-9\PRINT#P,TAB(13),C$((C1-1)*12+1,(C1+9)*12)
1050FORX=1TOR\PRINT#P,R$(12*(X-1)+1,12*X),\R1$=R$(12*(X-1)+1,12*X)
1060FORY=C1TOC1+9\GOSUB200\C1$=C$(12*(Y-1)+1,12*Y)\IFR1$=F$THEN1080
1070IFC1$=F$THEN1080\IFP$(1,2)="BL"THEN1080\IFC1$=F$THEN1080\PRINT#P,D(X,Y),\GOT
O1090
1080PRINT#P,F$,
1090NEXTY\PRINT#P\NEXTX\IFP=1THENPRINT#P,CHR$(12)\GOTO540
1100IFA$<>"S"THEN1150
1110INPUT1"Enter C1 ",C1\IFC1<2THEN1110
1120INPUT"Enter C2 ",C2\IFC2>CTHEN1120
1130C$((C1-1)*12-11,C2*12)=C$(C1*12-11,C2*12)+"            "
1140FORX=C1TOC2\FORY=1TOR\D(Y,X-1)=D(Y,X)\NEXT\NEXT\FORX=1TOR\D(X,C2)=0\NEXT\GOT
O540
1150IFA$<>"G"THEN1260\INPUT1"GRAPH - ENTER R1? ",R1\INPUT1"    C1? ",C1
1160INPUT1"    R2? ",R2\INPUT1"    C2? ",C2\N=R2-R1+C2-C1+1\IFN<=50THEN1180
1170INPUT"GRAPH TOO LARGE ",A$\GOTO540
1180T1=INT(50/N)\H=D(R1,C1)\L=D(R1,C1)\FORX=R1TOR2\FORY=C1TOC2
1190IFD(X,Y)>HTHENH=D(X,Y)\IFD(X,Y)<LTHENL=D(X,Y)\NEXTY\NEXTX\PRINT
1200INPUT"Enter Printer Number? ",P\IFP<>0THENINPUT"Enter Title? ",T1$
1210IFP<>0THENINPUT"Hit <CR> to continue...",A$\IFP=1THENPRINT#P,CHR$(28),
1220IFP=0THENL1=15ELSEL1=50\I=(H-L)/(L1)\PRINT#P,T1$\PRINT#P\PRINTCHR$(11)\FORX=
1TOL1
1230PRINT#P,H,\T=0\H=H-I\FORX1=R1TOR2\FORY1=C1TOC2
1240IFD(X1,Y1)>=HTHENPRINT#P,TAB(T*T1+14),"*",\T=T+1\NEXTY1\NEXTX1\PRINT#P\NEXTX
1250IFP=1THENPRINT#P,CHR$(12)\INPUT1"Hit <CR> to continue....",A$\GOTO540\GOTO54
0
1260IFA$<>"M"THEN1300\INPUT1"Mult. Entry - Enter R1 ",R1\INPUT1"Enter C1 ",C1
1270INPUT1"Enter R2 ",R2\INPUT1"Enter C2 ",C2\FORX=R1TOR2\FORY=C1TOC2\PRINTCHR$(
11)
1280PRINT\PRINT\PRINTTAB(14),C$((Y-1)*12+1,Y*12)\PRINT\PRINTR$((X-1)*12+1,X*12),
1290INPUTD(X,Y)\NEXT\NEXT\GOTO540
1300IFA$<>"I"THEN1330\FORX=1TOR\FORY=1TOC\GOSUB200\IFP$(1,2)<>"IP"THEN1320\PRINT
CHR$(11)\PRINT\PRINT
```

CON'T ON PAGE 8

CON'T FROM PAGE 7

```
1310PRINTTAB(14),C$((Y-1)*12+1,Y*12)\PRINT\PRINTR$((X-1)*12+1,X*12),\INPUTD(X,Y)
1320NEXT\NEXT\GOTO540
1330IFA$<>"A"THEN1890\PRINTCHR$(11)\PRINT\PRINT\PRINT\PRINTT$\PRINT\PRINT"ANALYS
IS",\FORY=1TOC\FORX=1TOR
1340GOSUB200\IFP$(1,2)="IP"THEN1880\IFP$(1,2)="  "THEN1880\IFP$(1,2)="BL"THEN188
0
1350GOSUB150\IFP$(1,2)<>"AD"THEN1360\D(X,Y)=D(V1,T1)+D(V2,T2)\GOTO1880
1360IFP$(1,2)<>"AA"THEN1380\D(X,Y)=0\FORX1=V1TOV2\FORY1=T1TOT2
1370D(X,Y)=D(X,Y)+D(X1,Y1)\NEXTY1\NEXTX1\GOTO1880
1380IFP$(1,2)<>"SU"THEN1390\D(X,Y)=D(V1,T1)-D(V2,T2)\GOTO1880
1390IFP$(1,2)<>"MD"THEN1400\D(X,Y)=D(V1,T1)*D(V2,T2)\GOTO1880
1400IFP$(1,2)<>"AK"THEN1420\D(X,Y)=D(V1,T1)+V3\GOTO1880
1410D(X,Y)=D(V1,T1)+V3\GOTO1880
1420IFP$(1,2)<>"SK"THEN1430\D(X,Y)=D(V1,T1)-V3\GOTO1880
1430IFP$(1,2)<>"MK"THEN1440\D(X,Y)=D(V1,T1)*V3\GOTO1880
1440IFP$(1,2)<>"DK"THEN1470\IFV3=0THEND(X,Y)=0\IFV3=0THEN1460
1450D(X,Y)=D(V1,T1)/V3
1460GOTO1880
1470IFP$(1,2)<>"ZE"THEN1480\D(X,Y)=0\GOTO1880
1480IFP$(1,2)<>"DD"THEN1510\IFD(V2,T2)=0THEND(X,Y)=0\IFD(V2,T2)=0THEN1500
1490D(X,Y)=D(V1,T1)/D(V2,T2)
1500GOTO1880
1510IFP$(1,2)<>"AB"THEN1520\D(X,Y)=ABS(D(C1,T1))\GOTO1880
1520IFP$(1,2)<>"XY"THEN1530\D(X,Y)=D(V1,T1)^D(V2,T2)\GOTO1880
1530IFP$(1,2)<>"YX"THEN1540\D(X,Y)=EXP((LOG(D(V1,T1)))/D(V2,T2))\GOTO1880
1540IFP$(1,2)<>"SN"THEN1560\D(X,Y)=D(V1,T1)*1.745*10^-2
1550D(X,Y)=SIN(D(X,Y))\GOTO1880
1560IFP$(1,2)<>"CO"THEN1580\D(X,Y)=D(V1,T1)*1.745*10^-2
1570D(X,Y)=COS(D(X,Y))\GOTO1880
1580IFP$(1,2)<>"AT"THEN1600\D(X,Y)=ATN(D(V1,T1))
1590D(X,Y)=D(X,Y)*5.7296*10^1\GOTO1880
1600IFP$(1,2)<>"AS"THEN1640\IFABS(D(X,Y))<>1THEN1620
1610IFABS(D(V1,T1))=1THEND(X,Y)=D(V1,T1)*(3.1415926/2)\GOTO1630
1620D(X,Y)=ATN(D(V1,T1)/SQRT(1-D(V1,T1)^2))
1630D(X,Y)=D(V1,T1)*5.7296*10^1\GOTO1880
1640IFP$(1,2)<>"AC"THEN1680\IFD(V1,T1)<>-1THEN1650\D(X,Y)=3.1415926\GOTO1670
1650IFD(V1,T1)<>0THEN1660\D(X,Y)=3.1415926/2\GOTO1670
1660D(X,Y)=ATN(SQRT(1-D(V1,T1)^2)/D(V1,T1))
1670D(X,Y)=D(X,Y)*5.7296*10^1\GOTO1880
1680IFP$(1,2)<>"SQ"THEN1690\D(X,Y)=SQRT(D(V1,T1))\GOTO1880
1690IFP$(1,2)<>"AV"THEN1710\A=0\D(X,Y)=0\FORX1=V1TOV2\FORY1=T1TOT2\A=A+1
1700D(X,Y)=D(X,Y)+D(XI,Y1)\NEXTY1\NEXTX1\D(X,Y)=D(X,Y)/A\GOTO1880
1710IFP$(1,2)<>"IN"THEN1740\P1=D(V2,T2)\P1=P1/V4\I=V3/100/V4
1720P=D(V1,T1)\IFP<=0THEN1880\I2=0\FORX1=1TOV4\I1=P*I\I2=I2+I1\P=P+J1-P1
1730IFP<=0THENP=0\IFP=0THENI=0\NEXTX1\D(X,Y)=I2\GOTO1880
1740IFP$(1,2)<>"GE"THEN1750\D(X,Y)=D(V1,T1)\GOTO1880
1750IFP$(1,2)<>"GI"THEN1760\D(X,Y)=0-D(V1,T1)\GOTO1880
1760IFP$(1,2)<>"PC"THEN1790\IFD(V2,T2)=0THEN1780
1770D(X,Y)=D(V1,T1)/D(V2,T2)*100\D(X,Y)=D(V1,T1)/D(V2,T2)*100
1780D(X,Y)=D(X,Y)-100\GOTO1880
1790IFP$(1,2)<>"PE"THEN1830\IFD(V2,T2)=0THEN1810
1800D(X,Y)=D(V1,T1)/D(V2,T2)*100
1810IFP$(1,2)="PE"THEN1820\D(X,Y)=D(X,Y)-100
1820GOTO1880
1830IFP$(1,2)<>"GA"THEN2290\A=0\A=0\N=0\IFT1<2THENT1=2\IFT2<T1THENTI=T1
1840FORX1=V1TOV2\FORY1=T1TOT2\IFD(X1,Y1-1)=0THEN1860
1850A=A+D(X1,Y1)/D(X1,YI-1)\N=N+1
1860NEXT\NEXT\IFN=0THEN1870\A=A/N\D(X,Y)=D(V2,T2)*A
1870GOTO1880
1880NEXTX\NEXTY\GOTO540\NEXTY
1890GOTO540\RETURN
1900END
1910A$=INCHAR$(252)\A=ASC(A$)\PRINTA,"     ",A$\GOTO1910\C1=1\R1=1
1920C1=C1\R1=R1
1930PRINTCHR$(14),\IFR1>R-9THENR1=R-9\IFR1<1THENR1=1\IFC1>C-3THENC1=C-3\IFC1<1TH
ENC1=1
1940PRINTT$\FORX=0TO3\PRINTTAB(12),%12I,C1+X,\NEXT\PRINT
1950PRINTTAB(15),C$(12*(C1-1)+1,12*(C1+3))\FORX=R1TOR1+9\PRINT%3I,X,
1960PRINTR$(12*(X-1)+1,X*12),\FORY=C1TOC1+3
1970!D$((X-1)*C*12+(Y-1)*12+1,(X-1)*C*12+(Y-1)*12+2)+F2$,\NEXTY\PRINT
1980NEXTX\PRINT\A$=INCHAR$(252)\A=ASC(A$)\IFA<>17THEN1990\C1=C1-1\GOTO1930
1990IFA<>1THEN2000\C1=C1+1\GOTO1930
2000IFA<>23THEN2010\R1=R1-1\GOTO1930
2010IFA<>26THEN2020\RI=R1+1\GOTO1930
2020IFA$<>"E"THEN2270\PRINT"Enter Parameter: ",\INPUT1"Row?  ",X\ERRSET2310,G1,G
2
2030INPUT1"   Column 1?  ",Y\INPUT1"   Column 2?  ",Z1\Z=Y
2040PRINTCHR$(13)+CHR$(13)+CHR$(23)+CHR$(13),\GOSUB140
2050PRINT"COMD: ",P$,\!%41,"  ",V1,"  ",T1,"  ",V2,"  ",T2,
2060PRINT%12F2,"  ",V3,"  ",V4
2070INPUT1"Enter Comd: ",A$\IFA$=""THENPRINT\IFA$=""THEN2260
2080IFLEN(A$)<>2THENPRINT"BAD COMMAND"\IFLEN(A$)=2THEN2100\FORZ=1TO1000\A$=A$\NE
XTZ
2090IFLEN(A$)<>2THEN2070
2100P$=A$+"              "\IFP$(1,2)="ZE"THEN2240\IFP$(1,2)="BL"THEN2240
2110IFP$(I,2)="IP"THEN2240\IFP$(1,2)="  "THEN2240\INPUT1" R1? ",V1
2120INPUT1" C1? ",G1\IFP$(1,2)="GE"THEN2240\IFP$(1,2)="GI"THEN2240
2130IFP$(1,2)="AB"THEN2240\IFP$(1,2)="AK"THEN2210\IFP$(1,2)="SK"THEN2210
2140IFP$(1,2)="MK"THEN2210\IFP$(1,2)="DK"THEN2210\IFP$(1,2)="SQ"THEN2210
2150IFP$(1,2)="SN"THEN2210\IFP$(1,2)="CO"THEN2210\IFP$(1,2)="AT"THEN2210
2160IFP$(1,2)="AS"THEN2210\IFP$(1,2)="AC"THEN2210\INPUT1" R2? ",V2
2170INPUT1" C2? ",G2\IFP$(1,2)="AV"THEN2240\IFP$(1,2)="AA"THEN2240
2180IFP$(1,2)="AD"THEN2240\IFP$(1,2)="SU"THEN2240\IFP$(1,2)="MD"THEN2240
2190IFP$(1,2)="DD"THEN2240\IFP$(1,2)="PE"THEN2240\IFP$(I,2)="PC"THEN2240
2200IFP$(1,2)="GA"THEN2240\IFP$(1,2)="XY"THEN2240\IFP$(1,2)="YX"THEN2240
2210INPUT1" VAR1? ",G3\IFP$(1,2)="AK"THEN2240\IFP$(1,2)="SK"THEN2240
2220IFP$(1,2)="MK"THEN2240\IFP$(1,2)="DK"THEN2240\IFP$(1,2)="GR"THEN2240
2230PRINT"VAR2? ",G4\PRINT
2240FORY=ZTOZ1\PRINT"Z=",Z,\PRINT"Z1= ",Z1,"Y=",Y\PRINT\PRINTP$,Y
2250T1=GI\T2=G2\V3=G3\V4=G4\GOSUB60\G1=G1+1\G2=G2+I\NEXTY
2260PRINTCHR$(14)\GOTO1930
2270IFA$<>"D"THEN2290\PRINT"DISPLAY: ",\INPUT1"Row ",R1\INPUT1"   Column ",C1
2280PRINTCHR$(1I),\GOTO1930
2290IFA<>0THEN2300\IFA=0THENRETURN
2300GOTO1930\!D$((X-1)*C*12+(Y-I)*12+1,(X-1)*C*12+(Y-1)*12+2)+F2$,
2310Z1=Y\GOTO2090
2320INPUT"Enter Number of Row to Delete",A3\R1$="            "
2330R$=R$(1,(A3-1)*12)+R$(A3*12+1,R*12)+R1$\FORX=A3TOR-1\FORY=1TOC
2340D(X,Y)=D(X+1,Y)\NEXTY\NEXTX\FORY=1TOC\D(R,Y)=0\NEXTY\FORX1=A3TOR-1\FORY=1TOC
2350X=X1+1\GOSUB140\P$=P$+F$\IFVI>=A3THENV1=V1-1\IFV2>=A3THENV2=V2-1\X=X1
2360GOSUB60\NEXTY\NEXTX1\FORY=1TOC\P$="  "+F$\V1=0\V2=0\V3=0\V4=0\T1=0\T2=0
2370X=R\GOSUB60\NEXTY\GOTO540
```

```
     TWELVE MONTH STOCK TREND - 1

               3/7/82   4/4/82   5/2/82   6/1/82   7/3/82   8/1/82   9/4/82  10/3/82  11/1/82  12/5/82
JOW JONES      807.36   838.57   848.36   819.54   796.99   808.60   923 11   907.76   991.72  1031.36
 12 MOS      10856.89 10700.46 10563.82 10418.36 10256.16 10112 42 10174.53 10221.56 10360.73 10499.49
 % CHANGE       99.84    97.22    94.49    92.33    90.33    88.80    89 98    90 83    92.66    94.11

ATT             56.38    55.75    54.38    52.88    50 25    52 25    56.50    57 63    59.25    60.25
 12 MOS        677.25   680.63   680.38   676.38   671.63   667.38   669.13   667 38   667 13   672.01
 % CHANGE      110.03   110.02   109.74   108.39   107.27   105.91   106.34   104.54   102 93   102.26

AVON            24.88    24.00    25.25    24.50    23.75    20.63    25.63    23.13    25.25    28.38
 12 MOS        409.38   394.88   381.38   365.38   351.00   335.13   325.64   314.89   307 27   303.15
 % CHANGE       95.43    91.06    87.35    82.64    78.94    75.69    73.88    71.93    70 74    70.19

BANK AM         19.63    18.50    18.75    16.63    17.38    17.38    17.63    18 25    22 50    23 00
 12 MOS        276.63   268.63   262.75   253.75   246.26   239.76   235.52   229.89   230.27   229.39
 % CHANGE       88.95    85.01    83.15    80.36    78.55    76.82    76.56    75.00    75.65    75 69

BETH STEEL      20.00    22.38    22.88    18.38    15.25    15.25    18.38    15.63    17.50    17.38
 12 MOS        281.13   274.25   268.88   262.50   253.13   244.50   240.63   235.89   232.26   227.52
 % CHANGE       95.34    90.85    86.91    83.93    80.55    78.12    77.50    77.34    77.45    77.22
```

CON'T ON PAGE 9

CON'T FROM PAGE 8

	3/7/82	4/4/82	5/2/82	6/1/82	7/3/82	8/1/82	9/4/82	10/3/82	11/3/82	12/5/82
BLACK DR	13.13	13.25	14.25	13 88	13.63	12.88	14.75	14 50	17.13	18.00
12 MOS	195.75	190.50	166.38	162.00	176.63	172.51	171.26	170.64	172.64	175.27
% CHANGE	85.15	83 69	81.88	80.31	77.66	76.88	78.34	79 64	82.06	84 31
BOEING	16.50	17.50	19.13	17.00	15.38	16.00	23.88	21.13	27.38	32.63
12 MOS	316.25	299.13	284.13	270.63	256.13	244.51	244.39	241.02	242.40	251.40
% CHANGE	71.61	67.05	63.67	61.14	58.78	57.53	59.53	60.63	62.55	67.42
CATERPILLAR	46.13	49.75	45.88	36.75	37.75	36.63	41.38	34.13	37.25	39.75
12 MOS	692.50	680.25	654.63	630.63	603.13	577.26	561.14	540.89	527.89	512.27
% CHANGE	104.31	101.13	93.62	88 88	83 55	79.28	76.58	74.08	72.99	71.42
DEERE	31.75	32.75	29 75	25.38	24.00	22.88	27.75	23.38	26.50	30.00
12 MOS	453.25	439 63	425.75	409.38	393.88	378.76	369.63	357.01	350.26	343.76
% CHANGE	98.11	92.46	86.95	82.06	77.86	74.50	72.58	70.99	70.74	71.02
DELTA AIR	30.38	31.75	29.50	30.75	32.50	29.75	32.75	27.88	31.63	40 00
12 MOS	658.13	620.38	579.38	528.63	492.75	458.38	436.13	402.76	380.39	368.14
% CHANGE	111.85	99.76	88.27	75.75	68.13	61.94	58 35	52.71	49.30	47.85
DOW CHEMICAL	20.88	23.63	22.75	21.50	20.88	20 25	24.63	23.25	25 75	25.50
12 MOS	344.00	331.50	319.88	307.63	296.76	285.76	283.51	281.01	280 89	279.01
% CHANGE	85 15	81 63	78 33	75.19	72.82	70.62	71.48	72.12	73.34	73.76
DUPONT	32.88	35.00	35.38	33.75	33.00	30.25	36.00	35.25	38 50	37.50
12 MOS	516.50	501.00	486.38	467.13	448.88	433.63	429.88	425.75	423.75	420.88
% CHANGE	102.30	96.63	91.64	86.01	81.32	78.17	78 16	78.05	77.57	77.03
KODAK	69.25	73.38	71.00	71.63	72.38	75.63	85 25	83 00	87.38	92.63
12 MOS	864.88	855.25	850.25	845.51	845.89	846.77	867.39	883.69	905.77	927.65
% CHANGE	113.56	107.01	103.22	100.06	98.50	96 95	99.30	101.68	104 52	106.58
EXXON	28.13	28.75	28.25	28.00	27.13	26.00	28.88	27.88	29.63	27.38
12 MOS	385.44	380.13	374.00	366.38	361.88	352 88	349 64	346.39	345.65	340.28
% CHANGE	88.80	87.30	85.10	83.61	82.30	80.30	80.26	81.03	82.33	82.49
FLUOR	19.88	20.00	18.75	17.50	16.88	13.50	17.00	17.13	21.13	21.25
12 MOS	392.13	367.63	344.88	328.13	309.88	287.01	268.13	255.01	247.39	238.64
% CHANGE	69.76	63.65	58.32	55.21	52.30	48.76	46.56	46.72	48.38	49 76
F.M.C.	25.00	26.00	27.25	25.38	24.88	25.38	29.25	31.13	31.75	32.00
12 MOS	344.25	335.50	330.13	325.00	317.51	311.89	313.64	318.77	324.64	328.89
% CHANGE	107.70	102.25	97.60	94.27	90.07	87.06	87.03	88.70	90.12	91.42
GEN ELEC	60.25	64.25	64.13	61.88	63.50	65.38	78.13	76.75	85.88	91 63
12 MOS	726.25	723.38	722.25	718.13	721.13	725.51	746.14	766.77	800.40	832.15
% CHANGE	108.86	105.76	102.68	99.91	99.62	99.27	102.17	104.65	108.93	113.14
GEN MOTORS	39.88	42.00	41.75	42.88	43.88	42.38	49.13	47.25	56.75	57.50
12 MOS	548.50	537.50	525.25	511.88	503.76	494.76	497.64	499.52	519.77	539.27
% CHANGE	95.45	93.15	89.65	85.60	83.68	81.66	83.39	84.40	89.73	93.87
GEN TEL	29.13	30.13	30.75	28.38	26.75	28.25	31.63	33.75	40.00	40 00
12 MOS	360.75	364.13	367.00	365.13	362.75	361.00	364.01	365.88	372.38	380.51
% CHANGE	111.77	112.47	112.92	111.15	110.13	108.69	108.98	108.13	108.21	108.79
GOODYEAR	21.50	22.00	23.13	22.38	24.38	23.25	26.63	25.00	29.38	34.00
12 MOS	227.63	229.88	234.25	238.88	244.26	248.01	256.01	263.14	274.64	290.02
% CHANGE	122.96	119.42	117.49	117.02	116.45	115.69	118.11	120.57	125.41	131.08
HARNISCHFEGE	9.00	9.50	8.88	7.75	7.25	6.75	7 50	8.75	7.50	6.63
12 MOS	154.50	147.13	140.63	134.50	128.00	120.00	114.63	112.38	107.75	101.88
% CHANGE	86.07	81.00	76.06	73.70	70.62	66.30	64.40	65.00	63.48	60.82
I.B.M.	58.50	61.50	64.25	61.50	60.00	65 63	72.13	74.13	79.88	88.25
12 MOS	665.50	685.75	690.88	694.00	697.13	706.63	724.76	743.89	772.27	805.90
% CHANGE	86.82	92.14	92.19	92.46	93.40	95.73	99.88	104.39	110.72	117.71

	3/6/82	4/4/82	5/1/82	6/5/82	7/3/82	8/1/82	9/4/82	10/3/82	11/3/82	12/5/82
I.T.T.	26 38	25.13	25.75	24.25	23.50	23.25	27.25	26.25	30.13	32.00
12 MOS	354.38	345.75	337.50	329.75	322.88	317.13	318.63	316.63	319.76	321.76
% CHANGE	102.24	97.70	93 49	90.25	87.86	86.20	87 84	87.98	89.32	89.84
MCDOND	31.00	34.75	37.75	34.25	35.50	37.63	43.75	38.50	41.25	39 50
12 MOS	404.13	396.13	397.00	391.38	388.00	389 51	405 01	416.88	426.63	433.63
% CHANGE	89.73	87.66	86.21	83.09	81.17	81.00	85 13	89.46	92.47	96.31
R	45.68	43.39	43.02	45.38	51.50	52.00	64 63	62.63	82.00	93 00
12 MOS	706 82	683.60	660.12	639.46	625.69	613.92	616.24	617.97	642.47	681.64
% CHANGE	89.48	84.49	81.68	79.26	77.77	76.65	77.44	78.30	82.44	89.02
RCA	18.56	18.53	18.96	18.50	16.63	17.75	21.38	22.50	23.88	23.63
12 MDS	285.67	275.38	265.70	256.20	245.97	237 81	234.59	233.88	236.02	238.96
% CHANGE	86 82	83.29	80.01	76.96	73.97	71.79	71.38	72.04	73.94	76 42
SEARS	16.84	16.54	17.16	18.75	18.88	19 38	23 00	22.88	28.63	30 88
12 MOS	208.81	209.24	209.94	211.68	213.02	214.44	219.22	223 76	234.29	247.32
% CHANGE	104.28	104.83	105.34	106.08	106.34	106.41	108 00	109.41	113.79	119 48
STD.O.IND	49.88	47.16	45.69	43.25	39.75	35.50	42.13	40.75	41.13	37.50
12 MOS	734.09	707.53	680.01	652.71	626.32	598.62	580 15	561 90	547.30	530.01
% CHANGE	88.22	84.53	80.82	77.42	74.53	71.76	70.20	68 81	66.12	67.15
TEXACO	33.56	32.58	31.97	29.50	28.75	26.75	29.75	29.25	30.63	29 63
12 MOS	441.83	433.00	423.54	412.16	401.74	394.06	387.38	380.81	376.33	371.13
% CHANGE	90.85	88.84	86.71	84.29	82.36	81.78	81.05	80.44	80.38	80.21
USS	27.80	27.02	26.52	21.25	18.38	17.75	19.25	17.88	18.88	16.75
12 MOS	357.24	354.47	351.20	342.66	330.52	317.35	305.40	292.69	281.88	271.26
% CHANGE	100.44	99.66	98.74	96.34	92.73	88.76	85.08	81.36	78.37	75.51
WELLSF	26.58	25.77	25.27	20.63	21.25	21.50	22.50	24.25	28.50	34.00
12 MOS	346.28	343.71	340.64	332.93	324.70	316.12	308.65	303.22	302.16	306 63
% CHANGE	101.82	101.07	100.16	97.90	95.16	92.17	89.57	87 65	87.04	88.03
WESTGEL	25.40	24.84	24.88	24.38	25.50	26.50	33.88	31.38	36.25	39.00
12 MOS	347.08	342.90	337.96	331.82	326.77	322.70	326.14	327.47	334.57	345 47
% CHANGE	105.91	104.04	101.71	98.85	96.36	94.21	94.30	93.89	95.38	98.21
XEROX	41.18	39.70	38.74	30.63	31.38	29.50	34.88	33.50	37.75	38.63
12 MOS	625.68	606.85	587.06	559.16	533.14	506.22	486.17	466.54	453.90	444.72
% CHANGE	89.08	86 40	83.58	79.61	76 03	72.41	69.90	67.60	66.56	66.25
MONEY MARKET	13.80	14.20	13.40	14.20	13.08	12.94	10.09	9.70	9.23	6.53
12 MOS	174.60	175.20	175.00	173.40	170.28	166.22	160.51	155.41	150.84	147.57
% CHANGE	106.85	107.22	107.10	104.71	101.24	96.66	92.35	88 80	86.09	85.10
GOLD	344.00	320.00	331.00	325.00	312 75	324.90	455 00	402.00	423.25	441.25
12 MOS	5130.00	4927.00	4772.00	4644.00	4542.75	4470.65	4483.65	4450.65	4444.90	4465.15
% CHANGE	72.00	69.00	67.00	66.98	67.96	69.41	71.90	74.10	76.70	79.75
SILVER	7.49	7.07	6.81	6.00	5.67	6.56	8.78	8.30	10 01	10.26
12 MOS	111.93	106.81	102.51	98.51	95.78	93.61	92 42	91.36	92.11	93 71
% CHANGE	58.00	56.00	54.00	53.96	54.98	56.35	57.76	61.60	66.57	72.39
S/G RATIO	45.92	45.26	48.60	54.17	55.16	49.53	51.82	48.43	42.28	43.01
12 MO %CHG	126.00	123.00	123.00	123.21	121.92	120.56	118.54	107.62	112.86	101.42

INSTRUCTIONS FOR RELOCATING ELECTRIC PENCIL I VERSION SS TO ANY PAGE BOUNDARY

by: John Osudar
P.O. Box 1451
Homewood, IL 60430-0451

NOTICE: The accuracy of these instructions is not guaranteed, and the author bears no liability for any errors contained within this document. (In other words, use at your own risk!) In particular, I have no idea how many revisions to Electric Pencil Version SS may exist, or which one I have. I only know that I have relocated MY copy of EP successfully to origins of 2D00H, 0100H, 0200H and 0300H. If you have any additional information, please send it to me at the above address.

WHAT IS REQUIRED

To do this, you need a code relocator. There are two of these available: one is in Leor Zolman's article, "A Machine Code Relocator for the 8080", in BYTE Volume 2 Number 7, July, 1977; pages 92-95. The other one, based on Leor Zolman's, is in an article by Joe Maguire in SOLUS NEWS (the predecessor of PROTEUS/NEWS, remember?!) Volume 1 Number 6, October/November, 1978; pages 10-12. If you have access to neither of these, a hex dump of Leor Zolman's program is supplied with these instructions.

WHAT YOU DO

(1) Load the ELECTRIC PENCIL program (from cassette or disk) into memory starting at its NEW destination. (e.g. if relocating to a new origin of 0300H, do a GET PENCL 300 from SOLOS, or the equivalent in your DOS.) Note that PENCL will occupy a total of 0E00H bytes of memory -- keep this in mind, so that you don't overwrite your DOS, or the relocator program, with the loaded PENCL (or vice versa.)

(2) Load the relocator program into a safe area of memory. (Safe = not occupied by operating system, PENCL, or other active code or data.)

(3) The relocator programs require six parameters, of which four remain unchanged throughout the relocation. For Leor Zolman's program, you must enter the parameters into the appropriate memory locations manually. Joe Maguire's version of the program prompts for the parameters and accepts input from the keyboard. In either case, the parameters required are (in order):

A. First address to be relocated
B. Last address to be relocated
C. Destination address of first byte
D. First address to have references fixed after move
E. Last address to have references fixed after move
F. Function code --
0 for fixing references only,
1 for moving and fixing references,
2 for moving only.

The relocator programs have the ability both to move (copy) the code from one place in memory to another, and to fix references; since you have loaded PENCL at its final destination already, you are only interested in using the reference-fixing function.
The parameters to be entered to relocate PENCL are as follows: (all values in hex; given first as a sixteen bit number that you would enter for Joe Maguire's program, then as the two eight bit values you would enter into memory for Leor Zolman's program):

A	0001	01	00
B	0FFF	FF	0F
C	xx01	01	xx
D	see table below		
E	see table below		
F	00	00	

Throughout this description, "xx" stands for the two hex digits representing the high digits of the new start address of PENCL, e.g. if PENCL is now starting at 0300H, then "xx" should be replaced by the hex number 03.

The following table gives the values to fill in for D and E above. EACH OF THESE NUMBERS MUST HAVE THE BASE ADDRESS "xx00" ADDED TO IT [OR "xx" ADDED TO THE SECOND (HIGH-ORDER) BYTE]! For example, if you are moving PENCL to an origin of 0300H, then the parameters for line 3 below are 04C2 (G2 04) and 052F (2F 05). The relocator program MUST be re-executed for EACH pair of D and E values!

	D values		E values	
1.	0000	(00 00)	004F	(4F 00)
2.	0053	(53 00)	01BF	(BF 01)
3.	01C2	(C2 01)	022F	(2F 02)
4.	0233	(33 02)	0257	(57 02)
5.	025B	(5B 02)	0508	(08 05)
6.	050C	(0C 05)	05A2	(A2 05)
7.	05A6	(A6 05)	05D3	(D3 05)
8.	05D7	(D7 05)	0705	(05 07)
9.	0709	(09 07)	0750	(50 07)
10.	075B	(5B 07)	07C6	(C6 07)
11.	07CA	(CA 07)	08BD	(BD 08)
12.	08C1	(C1 08)	0A13	(13 0A)
13.	0A17	(17 0A)	0C53	(53 0C)

(4) You must now make one change manually. At address 01C1+xx00 you should find a byte containing 0EH. Add the value xxH (i.e. the high byte of the base address) to it.

(5) Your Electric Pencil Version SS has now been relocated to a new origin of xx00H. Save it (on cassette or disk) and try executing it from SOLOS (with an EX xx00 command); it should operate just as it did when located at 0000H. If it doesn't, then either you have made a mistake during the relocation process, OR you have a different revision of PENCL than the one for which these instructions are given. If the latter is true, then I'm afraid that I can't help you.

NOTES

If you are planning on executing PENCL from an operating system other than SOLOS, you MUST set up the stack and registers similar to the way SOLOS does before jumping to PENCL. This is why I would make a CP/M version of PENCL start at 0300H -- I would place initialization code at 0100H, and end it with a JMP 0300H instruction or equivalent.
SOLOS jumps to an EXecuted program with HL pointing to the start of SOLOS itself (i.e. C000H), and the SP pointing to a useable stack area with the SOLOS return address on the stack already. You can emulate this by doing the following instruction sequence:

```
         LXI     SP,0CC00H           ;Set up to use SOLOS stack
         LXI     H,0C000H            ;Point to SOLOS start
         CALL    xx00H               ;Call PENCL
         JMP     0C004H              ;Upon return, jump to SOLOS
```

There is one additional problem with PENCL: it likes to grab all of available memory for its text buffer. It does this by scanning for the first byte of memory which it can't zero out (the first nonexistent memory or the first byte of read-only memory.) If you have your operating system sitting above PENCL somewhere (e.g. CP/M at the top of available memory) then PENCL will promptly and silently zero out your OS! This can be avoided simply enough: the code in PENCL that does the memory scan and initialization is located at xx2E through xx37, a total of ten bytes, and the last available address is stored at 0E08+xx00 (as usual, low byte followed by high byte.) You can replace the ten bytes with a CALL or JMP to a suitable patch routine, which will do the initialization properly for your configuration. (PENCL likes its buffer cleared to zeroes.)

CON'T ON PAGE 11

CON'T FROM PAGE 10

A word of warning: PENCL starts clearing memory at 0DF7H+xx00H, as should your patch routine. This, however, causes you to zero out the contents of PENCL's stack. PENCL doesn't have anything on the stack when it is doing the initialization; therefore, neither should your routine. (In particular, if you CALL the routine, the return address MUST be taken off the stack before doing the memory initialization.) Also, store the end address at 0E08H+xx00H AFTER you clear memory, for the same reason.
You will also need to patch a loop at 0A05H+xx00H through 0A0BH+xx00H, which is used by PENCL's CAA and CAB commands. Entry at 0A05H+xx00H is with the first address to be cleared contained in HL. PENCL does not appear to care what's in the registers when you're done, as long as the SP is still the same when you hit the RET instruction at 0A0CH+xx00H.

Please note that at this point you have only moved PENCL. PENCL will still NOT work with your disk operating system (i.e. it will not do any disk I/O -- it will still expect a cassette recorder to be used as the file I/O device.) Getting it to do that is far more complicated, but it can be done -- my PENCL uses my North Star DOS for disk I/O. The way I chose to implement this is based upon the fact that PENCL only knows the location of SOLOS by the value it receives in the HL register at the start. If you load a "fake SOLOS" into memory on some page boundary (e.g. I use F000H-FFFFH for a "fake SOLOS" in which the cassette I/O entries are replaced by N*DOS disk I/O) and load the start address of this "fake SOLOS" into HL before calling PENCL, then PENCL will use the entry points in your version of SOLOS. It uses the standard entry point jump table, so it's easy to put in disk I/O routines; you only have to emulate the cassette block I/O entry points RDBLK and WRBLK, and set up the other entries in the jump table to jump into the SOLOS routines in ROM. Remember that you MUST follow the SOLOS register conventions for the RDBLK and WRBLK routines!

LEOR ZOLMAN'S MACHINE CODE RELOCATOR FOR THE 8080

(This program was originally printed in the July, 1977 issue of BYTE magazine. It has been modified to jump to SOLOS at 0C004H instead of infinitely looping when done.)

Enter this program on a page boundary (i.e. zz00H)

The following is a hex dump of the program, sixteen bytes to a line. Note that "zz" represents the high byte of the starting address throughout.

```
31 FF zz 2A  DF zz 54 5D  2A E1 zz 44  4D 2A DD zz
C5 CD A9 zz  19 44 4D E1  09 3A E7 zz  B7 CA 3A zz
1A 77 78 B7  C2 2C zz 79  B7 CA 32 zz  2B 1B 0B C3
20 zz 3A E7  zz FE 02 CA  04 C0 E5 62  6B CD A9 zz
D1 19 22 E8  zz 2A E3 zz  2B 23 EB 2A  E5 zz EB 7B
95 7A 9C DA  04 C0 06 1A  11 E1 zz 1A  BE CA 7B zz
05 13 C2 5B  zz 06 12 11  CB zz 1A BE  CA 77 zz 05
13 C2 6A zz  C3 49 zz 23  C3 49 zz E5  2A DF zz 54
5D 2A DD zz  44 4D E1 23  7B 96 23 7A  9E DA 49 zz
2E 7E 91 23  7E 98 DA 49  zz 2B EB 2A  E8 zz EB 7E
83 77 23 7E  8A 77 C3 49  zz 7C 2F 67  7D 2F 6F 23
C9 01 11 21  22 2A 31 32  3A C2 C3 C4  CA CC CD D2
D4 DA DC E2  E4 EA EC F2  F4 FA FC 06  0E 16 1E 26
2E 36 3E C6  CE D3 D6 DB  DE E6 EE F6  FE
```

The next 11 bytes are used for parameter storage, as follows:

zzDD,zzDE:	First address of block to be relocated
zzDF,zzE0:	Last address of block to be relocated
zzE1,zzE2:	Destination address of first byte
zzE3,zzE4:	First address to have references fixed after move
zzE5,zzE6:	Last address to have references fixed
zzE7:	Function code: 00 = fix references only 01 = move block and fix references 02 = move block only

The two bytes at zzE8 and zzE9 are used internally for storage. Ordinarily, only the values at zzE3 through zzE6 should need to be modified when relocating parts of a large program (like ALS8 or PENCL.)

NOTES ON A CP/M INTERFACE FOR ELECTRIC PENCIL I VERSION SS

The attached 8080 assembly language program, PENCILIO, is a CP/M interface for Electric Pencil Version SS. It replaces the cassette I/O functions with CP/M disk I/O. Since CP/M is supposed to be "standard", this code should, in theory, work on any standard (0-origin) CP/M V2.x system. In light of Murphy's Law, however, nothing is guaranteed.

PENCILIO works by fooling PENCL into thinking that SOLOS is located at 0100H. This is done through the value passed to PENCL in HL when PENCL is first called. At 0100H we place a jump table laid out just like the one in SOLOS. The entry points that we are not changing are directly transferred to the SOLOS-in-ROM addresses (e.g. 0107H contains a JMP 0C007H.) The only entry points that are changed are: INIT, which does some special initialization; RETRN, which jumps to 0 to re-boot CP/M; and RDBLK/WRBLK, which do disk I/O instead of cassette tape I/O.

The functionality provided by PENCILIO is almost identical to that provided in the cassette version. Files are named with five characters internally (e.g. ABCDE); these names are translated to CP/M filenames by first replacing spaces and/or control characters with dashes ('-'), then extending the name to its full length by appending '-EP.TXT', so that ABCDE is stored on disk as ABCDE-EP.TXT. PENCILIO will rename an existing file of the same name to a .BAK file, allowing one level of backup. Read errors are treated as end-of-file, and write errors are ignored (except to terminate the operation); this is not good, but it's about all that can be done simply without changing PENCL itself. Units 1 and 2 are implemented to refer to A: and B:, but may be changed by modifying PENCILIO's MAKEFCB routines to select other units.

SUMMARY OF STEPS IN IMPLEMENTING ELECTRIC PENCIL I VERSION SS UNDER CP/M

```
(1) Relocate PENCL to 0300H
(2) SAVE 16 PENCIL.COM in CP/M
(3) Enter and assemble PENCILIO.ASM
(4) DDT PENCIL.COM
(5) In DDT, do:   IPENCILIO.HEX
                  R
     to read PENCILIO into memory
(6) Exit DDT with ^C or G0
(7) SAVE 16 PENCIL.COM again
(8) Now,   PENCIL   should get you into the CP/M PENCL!
```

NOTE: If you have moved your SOLOS and video memory to addresses other than 0C000H and 0CC00H respectively, then you will have to modify PENCILIO.ASM to point to the new SOLOS, and you will have to make the following change to the relocated PENCL:
At address 0DF4H+xx00H (i.e. at 10F4H if you have relocated PENCL to an origin of 300H) PENCL contains the address of the video memory, i.e. 00 CC -- you must change this to reflect the starting address of video memory in your system. PENCL will NOT work with the 24-by-80 modification (at least this version won't!)

CON'T ON PAGE 12

CON'T FROM PAGE 11

```
;
;        PENCILIO
;
;    (C) COPYRIGHT 1982 BY JOHN OSUDAR.  NO WARRANTY EXPRESSED
;        OR IMPLIED.  AUTHOR NOT LIABLE FOR ANY DAMAGES
;        RESULTING FROM THE USE OF THIS SOFTWARE.
;        PERMISSION TO COPY FOR NON-COMMERCIAL USE
;        GRANTED, PROVIDING THAT THIS COPYRIGHT NOTICE IS
;        REPRODUCED UNCHANGED AND IN ITS ENTIRETY.
;        NO PORTION OF THIS SOFTWARE MAY BE INCLUDED IN ANY
;        COMMERCIAL PRODUCT WITHOUT AUTHOR'S PRIOR CONSENT.
;
;        JOHN OSUDAR
;        POST OFFICE BOX 1451
;        HOMEWOOD, ILLINOIS 60430-0451
;
;        VERSION V01.06
;
;        WRITTEN 29-MAY-82 BY JOHN OSUDAR
;
;        THIS IS A PSEUDO-SOLOS THAT SUPPLIES THE RDBLK AND
;        WRBLK FUNCTIONS TO READ/WRITE CP/M FILES FROM THE
;        ELECTRIC PENCIL VERSION SS.  PENCL MUST BE LOCATED
;        AT ADDRESS 0300H OR ABOVE, AS THIS CODE OCCUPIES
;        ADDRESSES 0100H TO 02FFH
;
         ORG      100H             ;START AT TPA START
SOLOS    EQU      0C000H           ;START OF SOLOS IN ROM
CPM      EQU      5                ;ADDRESS FOR CP/M SYSTEM CALL
PENCIL   EQU      300H             ;START OF ELECTRIC PENCIL
REBOOT   EQU      0                ;CP/M REBOOT ADDRESS
;
;        PSOLOS
;
;        PSEUDO-SOLOS JUMP TABLE
;
;        EACH ENTRY POINT EXCEPT INIT, RETRN, RDBLK, AND WRBLK
;        IS DIRECTED TO THE CORRESPONDING SOLOS-IN-ROM ROUTINE
;
PSOLOS:
         DB       0                ;SHOWS THAT THIS IS SOLOS
         JMP      INITP            ;INITIALIZER
         JMP      REBOOT           ;RETURN (TO CP/M, NOT SOLOS)
         JMP      SOLOS+07H        ;FOPEN
         JMP      SOLOS+0AH        ;FCLOS
         JMP      SOLOS+0DH        ;RDBYT
         JMP      SOLOS+10H        ;WRBYT
         JMP      PRDBLK           ;CP/M RDBLK
         JMP      PWRBLK           ;CP/M WRBLK
         JMP      SOLOS+19H        ;SOUT
         JMP      SOLOS+1CH        ;AOUT
         JMP      SOLOS+1FH        ;SINP
         JMP      SOLOS+22H        ;AINP
;
;        INITP
;
;        INITIALIZATION CODE
;
INITP:
         LXI      SP,SOLOS+0C00H   ;SET UP STACK IN SOLOS AREA
         LXI      D,PTCH1COD       ;PREPARE TO PATCH PENCIL
         LXI      H,PENCIL+2EH     ;SET TO ADDRESS WHERE TO PATCH
         MVI      C,10             ;TEN BYTES PATCHED
PATCH1:  LDAX     D                ;COPY LOOP
         MOV      M,A              ;
         INX      D                ;
         INX      H                ;
         DCR      C                ;
         JNZ      PATCH1           ;
         LXI      D,PTCH2COD       ;SECOND PATCH
         LXI      H,PENCIL+0A06H   ;
         MVI      C,6              ;SIX BYTES PATCHED
PATCH2:  LDAX     D                ;COPY LOOP
         MOV      M,A              ;
         INX      D                ;
         INX      H                ;
         DCR      C                ;
         JNZ      PATCH2           ;
         LXI      H,PSOLOS         ;POINT TO OUR JUMP TABLE
         CALL     PENCIL           ;CALL ELECTRIC PENCIL
         JMP      REBOOT           ;REBOOT UPON RETURN
```

```
;
PTCH1COD:                          ;EXACTLY TEN BYTES OF CODE
         CALL     MEMPATCH         ;CALL IT
         NOP                       ;SOME NOP'S FOLLOW
         NOP                       ;
         NOP                       ;
         NOP                       ;
         NOP                       ;
         NOP                       ;
         NOP                       ;
;
PTCH2COD:                          ;EXACTLY SIX BYTES OF CODE
         CALL     ZERPATCH         ;
         NOP                       ;SOME NOP'S HERE TOO
         NOP                       ;
         NOP                       ;
;
MEMPATCH:                          ;PATCH ROUTINE
         POP      H                ;GET BACK RETURN ADDRESS
         SHLD     RETURN           ;STORE IT
         LHLD     0006H            ;GET CP/M START ADDRESS
         LXI      B,0FF7FH         ;SUBTRACT 81H
         DAD      B                ;
         SHLD     LAST             ;STORE AS LAST AVAILABLE ADDR
         LXI      D,0DF7H+PENCIL   ;START OF AREA TO BE ZEROED
INCLR:   MOV      A,H              ;NEGATE HL
         CMA                       ;
         MOV      H,A              ;
         MOV      A,L              ;
         CMA                       ;
         MOV      L,A              ;
         INX      H                ;
         XCHG                      ;SWAP POINTER & COMPARISON
CLRLOOP: MVI      M,0              ;CLEAR A BYTE
         INX      H                ;ADVANCE POINTER
         PUSH     H                ;SAVE IT
         DAD      D                ;ADD -HIGHEST ADDRESS
         POP      H                ;RESTORE
         JNC      CLRLOOP          ;LOOP UNTIL DONE
         LHLD     LAST             ;
         SHLD     PENCIL+0E08H     ;SET LAST ADDRESS
HERE:    JMP      0                ;RETURN FROM HERE
RETURN   EQU      HERE+1           ;STORE RETURN LOCATION HERE
LAST:    DS       2                ;
;
ZERPATCH:
         XGHG                      ;DE=START OF AREA TO BE ZEROED
         POP      H                ;GET RETURN ADDRESS
         SHLD     RETURN           ;STORE IT IN
         LHLD     PENCIL+0E08H     ;GET END ADDRESS
         JMP      INCLR            ;CLEAR IT

;
;        MAKEFCB1
;
;        MAKES .TXT FCB FROM SOLOS HEADER
;
MAKEFCB1:
         LXI      D,5CH            ;POINT TO FCB AREA
         ORA      A                ;TEST UNIT NUMBER
         JP       UNIT2            ;UNIT 2 IF HI BIT OFF
         MVI      A,1              ;ELSE UNIT 1
         JMP      DOUNIT           ;SET UNIT
UNIT2:   MVI      A,2              ;SELECT UNIT 2
DOUNIT:  STAX     D                ;SET UNIT IN FCB
         INX      D                ;ADVANCE
         MVI      C,5              ;LOOP COUNT
LOOP1:   MOV      A,M              ;COPY FILE NAME
         CPI      ' '+1            ;TRANSLATE SPACES/CONTROL
         JNC      NOSP             ; CHARS TO DASHES
         MVI      A,'-'            ;
NOSP:    STAX     D                ;STORE CHAR
         INX      H                ;ADVANCE
         INX      D                ;
         DCR      C                ;COUNT DOWN
         JNZ      LOOP1            ;AND LOOP
         XCHG                      ;SWAP POINTERS
         MVI      M,'-'            ;PUT IN '-EP.TXT'
         INX      H                ;
         MVI      M,'E'            ;
         INX      H                ;
         MVI      M,'P'            .
```

CON'T ON PAGE 13

CON'T FROM PAGE 12

```
                                        ;
         INX      H                     ;
         MVI      M,'T'                 ;
         INX      H                     ;
         MVI      M,'X'                 ;
         INX      H                     ;
         MVI      M,'T'                 ;
         INX      H                     ;
         XRA      A                     ;ZERO WHAT ELSE IS THERE
         MOV      M,A                   ;
         INX      H                     ;
         MOV      M,A                   ;
         XCHG                           ;SWAP POINTERS BACK
         RET                            ;DONE
;
;        MAKEFCB2
;
;        MAKES .BAK FCB FROM HEADER, THEN CALLS MAKEFCB1
;
MAKEFCB2:
         PUSH     PSW                   ;SAVE FOR MAKEFCB1
         PUSH     H                     ;
         LXI      D,6CH                 ;LIKE ABOVE, BUT 2ND FCB
         ORA      A                     ;
         JP       AUNIT2                ;
         MVI      A,1                   ;
         JMP      DOAUNIT               ;
AUNIT2:  MVI      A,2                   ;
DOAUNIT:  STAX    D                     ;
         INX      D                     ;
         MVI      C,5                   ;
ALOOP1:  MOV      A,M                   ;
         CPI      ' '+1                 ;
         JNC      ANOSP                 ;
         MVI      A,'-'                 ;
ANOSP:   STAX     D                     ;
         INX      H                     ;
         INX      D                     ;
         DCR      C                     ;
         JNZ      ALOOP1                ;
         XCHG                           ;
         MVI      M,'-'                 ;PUT IN '-EP.BAK' HERE
         INX      H                     ;
         MVI      M,'E'                 ;
         INX      H                     ;
         MVI      M,'P'                 ;
         INX      H                     ;
         MVI      M,'B'                 ;
         INX      H                     ;
         MVI      M,'A'                 ;
         INX      H                     ;
         MVI      M,'K'                 ;
         INX      H                     ;
         XRA      A                     ;
         MOV      M,A                   ;
         INX      H                     ;
         MOV      M,A                   ;
         POP      H                     ;RESTORE FOR MAKEFCB1
         POP      PSW                   ;
         JMP      MAKEFCB1              ;CALL MAKEFCB1 AND RETURN
;
;        PRDBLK
;
;        CP/M READ BLOCK ROUTINE
;
PRDBLK:
         PUSH     D                     ;SAVE ADDRESS OF BUFFER
         CALL     MAKEFCB1              ;MAKE FCB
         MVI      C,0FH                 ;OPEN FILE
         LXI      D,5CH                 ;
         CALL     CPM                   ;
         INR      A                     ;FAILED?
         POP      D                     ;RESTORE POINTER
         RC                             ;IF FAILED TO OPEN, RETURN
         XRA      A                     ;SET RECORD 0
         STA      7CH                   ;
RDLOOP:  MVI      C,1AH                 ;SET DMA ADDRESS
         PUSH     D                     ;SAVE POINTER
         CALL     CPM                   ;SET IT
         MVI      C,14H                 ;READ SEQUENTIAL
```

```
         LXI      D,5CH                 ;
         CALL     CPM                   ;
         ORA      A                     ;SUCCESS?
         POP      D                     ;RESTORE FIRST
         JNZ      RDDONE                ;IF FAILED, ASSUME EOF
         LXI      H,80H                 ;ELSE ADVANCE ADDRESS
         DAD      D                     ;
         XCHG                           ;
         JMP      RDLOOP                ;AND LOOP
RDDONE:  XCHG                           ;PUT END POINTER INTO HL
NOTEND:  MOV      A,M                   ;
         ORA      A                     ;
         DCX      H                     ;
         JZ       NOTEND                ;LOOK FOR FIRST ZERO BYTE
         INX      H                     ;POINT TO BYTE AFTER IT
         INX      H                     ;
         INX      H                     ;
         RET                            ;
;
;        PWRBLK
;
;        CP/M WRITE BLOCK ROUTINE
;
PWRBLK:
         CALL     MAKEFCB2              ;MAKE .TXT AND .BAK FCB'S
         INX      H                     ;ADVANCE IN HEADER TO SIZE
         INX      H                     ;
         MOV      C,M                   ;GET SIZE INTO BC
         INX      H                     ;
         MOV      B,M                   ;
         INX      H                     ;
         MOV      E,M                   ;GET ADDRESS INTO DE
         INX      H                     ;
         MOV      D,M                   ;
         LXI      H,7FH                 ;ROUND SIZE UP TO MULTIPLE
         DAD      B                     ; OF 80H
         MOV      A,L                   ;
         ANI      80H                   ;
         MOV      L,A                   ;
         PUSH     H                     ;SAVE SIZE AND POINTER
         PUSH     D                     ;
         LXI      D,6CH                 ;DELETE OLD .BAK FILE
         MVI      C,13H                 ;
         CALL     CPM                   ;
         LXI      D,5CH                 ;RENAME .TXT TO .BAK
         MVI      C,17H                 ;
         CALL     CPM                   ;
         LXI      D,5CH                 ;MAKE NEW .TXT FILE
         MVI      C,16H                 ;
         CALL     CPM                   ;
         INR      A                     ;SUCCESS?
         POP      D                     ;RESTORE SIZE AND POINTER
         POP      H                     ;
         RC                             ;IF FAILED, RETURN
         XRA      A                     ;SET RECORD 0
         STA      7CH                   ;
WRLOOP:  PUSH     H                     ;ELSE RESAVE SIZE AND POINTER
         PUSH     D                     ;
         MVI      C,1AH                 ;SET DMA ADDRESS
         CALL     CPM                   ;
         LXI      D,5CH                 ;WRITE SEQUENTIAL
         MVI      C,15H                 ;
         CALL     CPM                   ;
         POP      D                     ;RESTORE SIZE AND POINTER
         POP      H                     ;
         ORA      A                     ;SUCCESSFUL WRITE?
         JNZ      DOCLOSE               ;IF NOT, CLOSE
         LXI      B,0FF80H              ;ELSE COUNT DOWN SIZE
         DAD      B                     ;
         MOV      A,L                   ;TEST IF ZERO YET
         ORA      H                     ;
         JZ       DOCLOSE               ;IF SO, CLOSE FILE
         XCHG                           ;SWAP SIZE AND POINTER
         LXI      B,80H                 ;ADVANCE POINTER
         DAD      B                     ;
         XCHG                           ;SWAP THEM BACK AGAIN
         JMP      WRLOOP                ;LOOP
DOCLOSE:  MVI     C,10H                 ;CLOSE FILE
         LXI      D,5CH                 ;
         CALL     CPM                   ;
         RET                            ;DONE

         END
```

WHERE

A SUBROUTINE TO LOCATE SOLOS

Leonard Morgenstern

Dear Stan,

I have recently installed my new SOLOS, which will run either at F000 or C000. The job was done by the Computer Service Center, 1514 University Ave., Berkeley (415-845-6518). They charged me 3 hours at their current rate. The job included cleaning up the interior, etc. Since previous attempts on my part to deal with hardware have been disasters, I feel that I got my money's worth.

To assist me in reprogramming the I/O, I wrote a subroutine that will locate SOLOS, called WHERE. WHERE works by attempting to increment F000. If F000 can be incremented, then it is RAM, and SOLOS must be at C000. If F000 cannot be incremented, then there are two possibilities. There may be no memory there, in which case it will contain the value FF; or SOLOS may be there, in which case F000 will contain 0. WHERE sets the non-zero flag if SOLOS is at F000, and sets the zero flag if it is at C000.

I have incorporated WHERE in the initialization routine of my North Star DOS (see listing). The routine sets up an I/O jump table and also sets the DOS page length byte at DOS+33H. As written, the program will abort if SOLOS is not at C000. By deleting the last few commands, the North Star DOS becomes indifferent to the location of SOLOS.

```
* WHERE
*
* A SUBROUTINE TO LOCATE SOLOS
*
* Leonard Morgenstern
* 304 Rheem Blvd.
* Moraga CA, 94556
* December 1, 1982
*
* PERSONALIZATION
DOS     EQU     0D000H  Set to location of DOS
*
***************************************************
* WHERE tests location of SOLOS. Returns Z if at  *
* F000, NZ if at C000                             *
***************************************************
WHERE   LXI     H,0F000H  Set HL=F000
        INR     M       Try to increment F000
        MOV     A,M     Result to A-register
        DCR     M       Restore F000 to original
        CMP     M       Compare A with F000
        RNZ     ,       If unequal then SOLOS is
*                        not at F000
        ORA     A       Does F000 EQUAL 0?
        RET     ,       If Z then SOLOS is at F000; if
*                        NZ then it's at C000
*
***************************************************
*  Jump table for SOLOS I/O. Will be reset to     *
*  correct addresses by TINIT. NORTH STAR I/O     *
*  words such as CIN, COUT, ETC. must use these   *
*  not the SOLOS PROM.                            *
***************************************************
SOLOS   JMP     00004H
MSGV    JMP     00007H
AOUT    JMP     0001CH
AINP    JMP     00022H   (Others can be added)
*
***************************************************
* SBYTE sets A=F0 or C0, according to location    *
*   of SOLOS. It sets B-register to correct page  *
*   length, that is 16 decimal for SOLOS at C000  *
*   and 24 decimal for SOLOS at F000.             *
***************************************************
SBYTE   CALL    WHERE
        MVI     A,0C0H
        MVI     B,16
        RNZ
        MVI     A,0F0H
        MVI     B,24
        RET
*
***************************************************
* TINIT is Northstar DOS initialization routine   *
***************************************************
TINIT   CALL    SBYTE   A-register=F0 or C0
        STA     SOLOS+2  Patch jump table
        STA     AOUT+2
        STA     AINP+2
        STA     MSGV+2
        PUSH    PSW
        MOV     A,B
        STA     DOS+33H  Set page length byte
        POP     PSW
*
***************************************************
*   Next 5 instructions abort if SOLOS is         *
*   not at C000. Replace with a RET command to    *
*   make DOS "transparent" to the location of     *
*   SOLOS                                         *
***************************************************
        CPI     0C0H
        RZ
        LXI     H,MSG1
        CALL    MSGV
        JMP     SOLOS   JMP TO SOLOS ON ERROR
*
MSG1    DB      0DH
        ASC     'SOLOS NOT AT C000'
        DB      0DH
        DB      0
*
*
```

METIER

VERSION 93 MONITOR PROM
by Rick Downs.

PAGE 1

December 25, 1982

Dear Stan:

Happy Holidays from Denver, CO. We just had the storm of the century. It snowed for a solid 24 hours and set a new record for the most snow in a single 24 hour period - 24". Since everyone is snowed in and my relatives could not arrive for Christmas why not get caught up on some letter writing.

Since my last letter my SOL has undergone some dramatic changes - It's ready to face the battle of the IBM-PC. No it does not have a 16 bit CPU yet, but for myself the changes have been worth every dollar spent. I won't trade my SOL for anything. It's still, in my opinion, a fine machine, and with CP/M running it is certainly capable of holding it's own in the small computer arena.

After all the reports of Bob Hogg's video upgrade I knew it was a matter of time before I had to have one. While on business in California before Thanksgiving I had the pleasure of meeting Bob in person and seeing first hand the operation he is running down in Mission Viejo. I allowed myself 1 hour to spend with him, and ended up staying 3. I walked out with one Dual 80/64 video board and a new BMC green phospher monitor. Bob, as one person said, will be the savior of the SOL. He has more SOL hardware sitting around unlike anything I have seen before. I think I can safely say that Bob definitely knows the machine(SOL) inside and out.

Another item I purchased while at Bobs was the new monitor PROMS necessary to use the Dual 80/64 board in the SOL. Since I had bought the Dual Personality earlier in the year I only needed the PROMS. Bob has added some enhancements to the PROMS to provide some extra features. They are listed below:

VERSION 93 MONITOR PROM

(1) ^O - Turns on inverse video

(2) ^T - Turns off inverse video

(3) ^Shift Del - Clears to end of line

(4) ^^ - Sets mode: not to clear to end of line on 'return'.

(5) Several other monitor functions have been modified for ease of use.

After arriving home from my trip to CA. I spent the better part of Thanksgiving day installing, and testing out the new video board. Everything that had been said was true - the improvement was incredible. Of course the new monitor played a part as well, in the quality of the display. It would be difficult to go back to a black/white video display although I still revert to the 64 column mode for some software that I haven't modified yet, and to use Electric Pencil. I am very pleased with the performance of the Dual 80/64 video board, and look forward to more enhancement for the SOL from Micro Complex.

On another note - let me say that for those SOL users who are about to buy SuperCalc, that the North Star format disk version is written in Z80 code. It will not run on the standard SOL. The solution to this dilema is to purchase the standard 8" version - it is written in 8080 code and therefore will run on the SOL. However it can't unless you've discovered how to stuff an 8" disk into a 5 1/4" drive and make it work. The answer is to get someone you know that has both an 8" system and North Star disk to download it for you. Also SuperCalc works best with an 80x24 screen format and since I upgraded I needed to have it modified for the Dual 80/64 video board. While at Bob's he gave me the name of an individual who will 'install' SuperCalc on the 80/64 board. His name is Dan Hunt of Costa Mesa, CA, and his phone number is (714) 549-8673. However, there is a price, but Dans' rates are very reasonsible. He modified my copy and it works like charm. I'm still learning to use this powerful software - but have been pleased with the results to-date. Dan will also modify Wordstar for the 80x24 screen, which allows more information to be displayed on some of the Wordstar menus.

Last summer, after getting CP/M up and running I was unhappy with the user area that comes standard from Lifeboat Associates and wanted something that would allow more control of such things as printer drivers, video display etc. I was use to my North Star DOS that had a user area written by Fr. Tom McCahee and thought - why not a good, functional user area for CP/M. I approached Fr. Tom with my specs for a good user area and he agreed to write one for me. With my ideas Fr. Tom added some of his own features and together a SUPER User Area was born. I paid alittle more than usual for the user area but have been very happy with it. After some discussion on the phone with Fr. Tom we agreed that he would make it available to other SOL users at minimal cost. The write up of the specs should appear in this issue with ordering and cost information from Fr. Tom McGhage, but if not I will list some of the more useful features.

SUPER-USER AREA FOR LIFEBOAT CP/M

(1) Screen Size - 64x16 or 80x24

(2) Page Length - number of printed lines/page

(3) New Page - automatic formfeed if desired

(4) SOLOS - C000 or F000

(5) Video Toggle - List 16 lines and stops, or as many lines as you set.

(6) Printer Toggle - Directs printer staus ON/OFF Inverse video 1 displayed if active.

(7) Paper - Sheet or Fanfold, this can be changed under program control

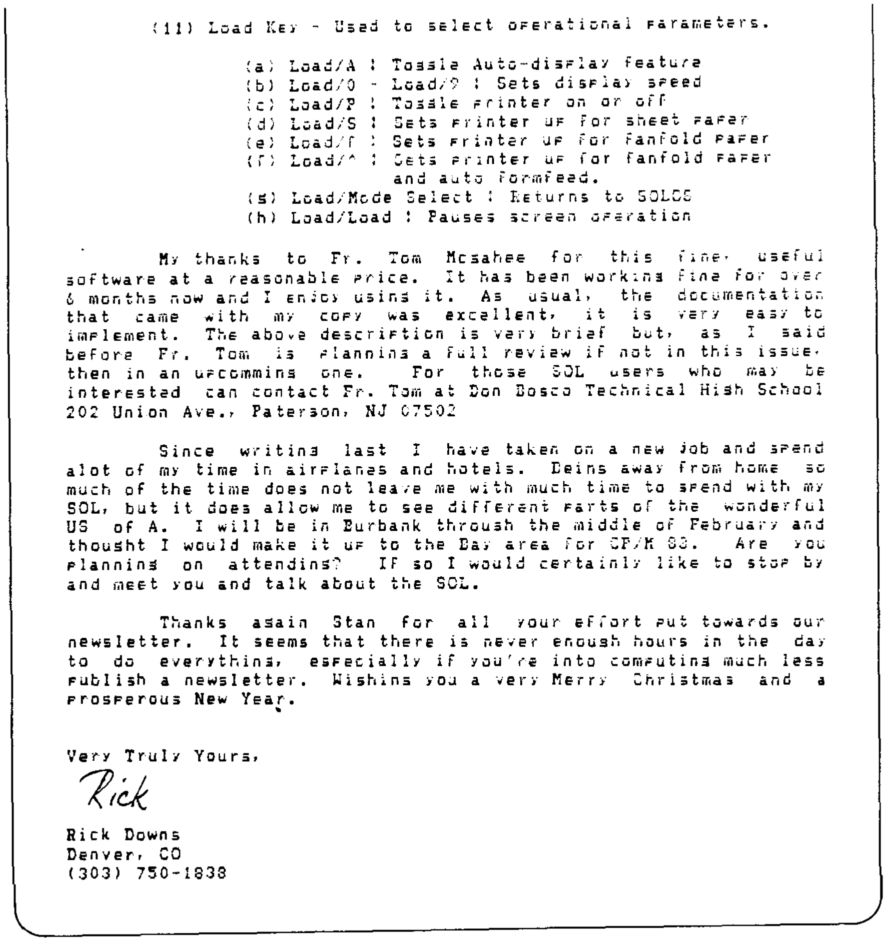
(8) Mode Select Key - Return to SOLOS

(9) Left Arrow Key - Converted to backspace

(10) Delete Key - Converted to backspace as well

CON'T ON PAGE 16

CON'T FROM PAGE 15



(11) Load Key - Used to select operational parameters.

(a) Load/A : Toggle Auto-display feature
(b) Load/0 - Load/9 : Sets display speed
(c) Load/P : Toggle printer on or off
(d) Load/S : Sets printer up for sheet paper
(e) Load/f : Sets printer up for fanfold paper
(f) Load/^ : Sets printer up for fanfold paper
and auto formfeed.
(g) Load/Mode Select : Returns to SOLOS
(h) Load/Load : Pauses screen operation

My thanks to Fr. Tom Mcgahee for this fine, useful software at a reasonable price. It has been working fine for over 6 months now and I enjoy using it. As usual, the documentation that came with my copy was excellent, it is very easy to implement. The above description is very brief but, as I said before Fr. Tom is planning a full review if not in this issue, then in an upcomming one. For those SOL users who may be interested can contact Fr. Tom at Don Bosco Technical High School 202 Union Ave., Paterson, NJ 07502

Since writing last I have taken on a new job and spend alot of my time in airplanes and hotels. Being away from home so much of the time does not leave me with much time to spend with my SOL, but it does allow me to see different parts of the wonderful US of A. I will be in Burbank through the middle of February and thought I would make it up to the Bay area for CP/M 83. Are you planning on attending? If so I would certainly like to stop by and meet you and talk about the SOL.

Thanks again Stan for all your effort put towards our newsletter. It seems that there is never enough hours in the day to do everything, especially if you're into computing much less publish a newsletter. Wishing you a very Merry Christmas and a prosperous New Year.

Very Truly Yours,

Rick

Rick Downs
Denver, CO
(303) 750-1838

Book Review

PASCAL IMPLEMENTATION: THE P4 COMPILER

Reviewed by Stan Sokolow

I came across this book among those offered in a direct mail packet from Byte magazine. It was rather expensive, about $60 as I recall, but it is every bit worth its price to someone who wants to fully understand the Pascal compiler known as the P4 compiler.

The P4 compiler is the one which originated in Zurich, written by Urs Ammann, Kesav Nori, and Christian Jacobi. It became the basis of the UCSD Pascal compiler, as well as many others. This is the compiler that coined the name "p-code" for its pseudo-machine's language.

The SLAC Pascal compiler we have in the Helios disk library is an improved version of the P4 compiler. So if you want to understand how to modify our SLAC Pascal compiler, this book will be essential.

Actually, the book comes in two volumes, both rather thin. Volume 1 contains 172 pages of explanatory text which walks through the compiler, assembler, and interpreter in great detail. Volume 2 contains 82 pages of source listings for the programs in Pascal. Every routine is explained. Variable names and their meanings are defined. Algorithms and data structures are explained and diagrammed. The p-codes are defined. Errors are pointed out. (Yes, the originally distributed P4 had some flaws as wells as vestiges of prior code that had been changed.) Line numbers in the listings are referenced in the explanation so you can follow along in the program. Some improvements are suggested.

Appendices give valuable cross references to the source code, such as a listing of all routine headings and their line numbers so you can get a quick index to the procedure and function names, their arguments, and their location in the listing. Error numbers are all listed. Cross reference tables are given for all of the identifiers used.

The introduction gives a brief overview of the compilation process and suggests sources for additional reading.

The explanations are not suitable for the total novice to compiler construction, but if you have some knowledge about the inner workings of compilers (or if you follow up on the background reading suggested), this text will give you what you need to master the P4 compiler and the p-machine interpreter. It is also an excellent way to learn about advanced techniques in the Pascal language.

Pascal Implementation: The P4 Compiler, by S. Pemberton and M.C. Daniels. Ellis Horwood Publishers, 1982. Distributed in North America by Halsted Press division of John Wiley & Sons, 605 Third Avenue, New York, N.Y. 10016, USA. Distributed in Canada by John Wiley & Sons Canada Ltd, 22 Worcester Road, Rexdale, Ontario, Canada. Distributed in other countries by John Wiley & Sons. The publishers will send you the book on approval.

CONVERTING 32KRA-1 TO 64KRA-1
by Leon Winter

Jan 11, '83

Dear Stan,

Enclosed with this letter you will find a copy of a modified switch position table for the 64/48 KRA-1. This represents part of the most receint work I've been doing with the Sol.

The table also has information for burning new PROMs (U37, U38, U39 & U40) for the addressing logic to make an N KRA-1 into a 64 KRA-1. Four of these PROMs are required (3 for a 48 KRA-1) to make the board a full 64K (48K). All four output lines should be programed low at the address locations given. All other locations are to remain high.

My origional Sol came with an N KRA-1 Rev C configured as a 32 KRA-1. This board was in turn only half populated with RAMs and had 2 of the 4 address PROMs missing. So I really only had 16K. Then, about 2 years ago, I started to work on making this card a full 32K. I first had to find out what was in the PROMs. That was easy on a logic breadboard. I next had 2 PROMs burned to the same program. The origional PROMs were 74S387s which require pull up resistors on the 4 output lines. At the time I could not get these so I used 74S287s which do not need external pull ups. This board has never even slightly noticed that 2 of its 4 address PROMs have different output configureations. So I left the external pull ups on the card for all 4 address PROM locations.

The original dynamic RAMs were Mostek MK4108s which are 8K fall outs from the 16K by 1 manufacturing process. These 8K chips were really 16Ks, but the upper or lower half failed the makers quality control tests. The MK4108s were no longer available so I bought sixteen Mostek MK4116s (16K by 1) to fill out my board. Thus I had a 32K memory for just a little cost.

Now you would think that I would be happy. Well greed soon struck. I kept thinking about the fact that half of the RAMs on my memory board could in themselves give me 32K! If only I could address the other half of each chip. Of course this would require a new set of address PROMs with a different program. I considered what that program must be like. That is the listing I included with this letter. The program is the same as that required for the 48 KRA-1, but only 3 PROMs are used. You will notice that the table is not exactly in a normal list form. I made it like the DIP Switch address select table. Then, because the actual PROM address is inverse of that selected by switch (pull ups on the input), I inverted the table. A moments looking will make it apparent.

HELP! Please. I'm allready to go on making this a 64K card, but I have 2 missing pieces of data. The origional manual I had for the 32 KRA-1 included four drawings for a plug-in header called a configuration module. This allowed PTC to use different dynamic RAMs and also to take care of addressing the RAMs. ei. A6 could be used for Col 0 or someother purpose, etc. With this module, the use of the upper or lower half of the RAM or both was controled. I was in hopes that Volume 6 (memory boards) of the Encyclopedia Porcessor Technica would have this info, but my hopes were dashed as looked through it. My first request of the membership is that someone send me a copy of the diagrams for the configuration modules for the 64/48 KRA-1. Since these were missing from the EPT for the 32 KRA-1, I will send a Xerox of that page. This may help someone else.

The last thing I need is a listing of the 64/48 KRA-1 program in the 74S278 PROM (U56). This is called the STATE MACHINE. I made a listing of the one for the 32 KRA-1, but I don't know if it's the same in for the 64K version. So a copy of this will be most valuable. If someone takes the trouble to run this out on a bread board for me, it would be good if they sent you a copy too, Stan.

I will close this out for now. If I'm not mistaken, Proteus News dues should be due about now. I don't know what the new rate is through. Wishing you the best in the new year.

Sincerely,

H Leon Winter

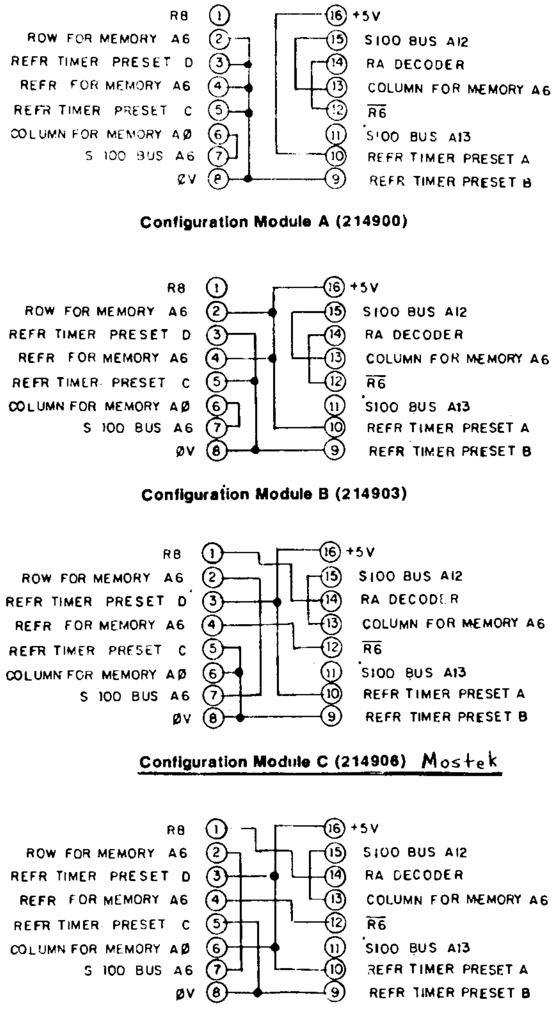
H LEON WINTER
1794 E MADISON AVE
EL CAJON, CA 92021
USA

N KRA-1	64/48 KRA-1		PROM ADDR BITS	
DIP SWITCH	16/4K BLOCKS		HEX	BINARY
POSITIONS	START	END		7654 3210
	F000	2FFF		
15...12	8,192	12,287	02	0000 0010
	4,096	8,191	01	0000 0001
	000	4,095	00	0000 0000
1111	61,440	65,535	0F	0000 1111
	E000	1FFF		
	4,096	8,191	11	0001 0001
	000	4,095	10	0001 0000
	61,440	65,535	1F	0001 1111
1110	57,344	61,439	1E	0001 1110
	D000	0FFF		
	000	4,095	20	0010 0000
	61,440	65,535	2F	0010 1111
	57,344	61,439	2E	0010 1110
1101	53,248	57,343	2D	0010 1101
	C000	FFFF		
	61,440	65,535	3F	0011 1111
	57,344	61,439	3E	0011 1110
	53,248	57,343	3D	0011 1101
1100	49,152	53,247	3C	0011 1100
	B000	EFFF		
	57,344	61,439	4E	0100 1110
	53,248	57,343	4D	0100 1101
	49,152	53,247	4C	0100 1100
1011	45,056	49,151	4B	0100 1011
	A000	DFFF		
	53,248	57,343	5D	0101 1101
	49,152	53,247	5C	0101 1100
	45,056	49,151	5B	0101 1011
1010	40,960	45,055	5A	0101 1010
	9000	CFFF		
	49,152	53,247	6C	0110 1100
	45,056	49,151	6B	0110 1011
	40,960	45,055	6A	0110 1010
1001	36,864	40,959	69	0110 1001
	8000	BFFF		
	45,056	49,151	7B	0111 1011
	40,960	45,055	7A	0111 1010
	36,864	40,959	79	0111 1001
1000	32,768	36,863	78	0111 1000
	7000	AFFF		
	40,960	45,055	8A	1000 1010
	36,864	40,959	89	1000 1001
	32,768	36,863	88	1000 1000
0111	28,672	32,767	87	1000 0111
	6000	9FFF		
	36,864	40,959	99	1001 1001
	32,768	36,863	98	1001 1000
	28,672	32,767	97	1001 0111
0110	24,576	28,671	96	1001 0110
	5000	8FFF		
	32,768	36,863	A8	1010 1000
	28,672	32,767	A7	1010 0111
	24,576	28,671	A6	1010 0110
0101	20,480	24,575	A5	1010 0101
	4000	7FFF		
	28,672	32,767	B7	1011 0111
	24,576	28,671	B6	1011 0110
	20,480	24,575	B5	1011 0101
0100	16,384	20,479	B4	1011 0100
	3000	6FFF		
	24,576	28,671	C6	1100 0110
	20,480	24,575	C5	1100 0101
	16,384	20,479	C4	1100 0100
0011	12,288	16,383	C3	1100 0011
	2000	5FFF		
	20,480	24,575	D5	1101 0101
	16,384	20,479	D4	1101 0100
	12,288	16,383	D3	1101 0011
0010	8,192	12,287	D2	1101 0010
	1000	4FFF		
	16,384	20,479	E4	1110 0100
	12,288	16,383	E3	1110 0011
	8,192	12,287	E2	1110 0010
0001	4,096	8,191	E1	1110 0001
	0000	3FFF		
	12,388	16,383	F3	1111 0011
	8,192	12,387	F2	1111 0010
	4,096	8,191	F1	1111 0001
0000	000	4,095	F0	1111 0000

CON'T ON PAGE 18

CON'T FROM PAGE 17

X-32



Configuration Module A (214900)

Configuration Module B (214903)

Configuration Module C (214906) Mostek

Configuration Module D (214909)

Fig. 7-5. 32KRA-1 Configuration Module Wiring Diagrams

7.5

A REVIEW OF PEARL-III

by Earl Dunham

Dear Stan;

Here's my impressions of Pearl-III as you requested. We have had Pearl for about a year and a half. For the money, it is one of the best of the few really helpful packages available for Sol, especially if one has accounting or data-base management needs in a small business or personal finances.

It should be noted that Pearl does not have much value for program developement outside of financial and related fields. For games, scientific applications, casual small programs for aid in programming, and anything that one can do without to much hassle, Pearl is not the way to go. But for inventory control, data-base management that does not warrant the cost of dBase II or the like, accounting (if one is an accountant, others should leave that to the pros), mailing lists, etc., it can be a real time saver.

A lot of time and thought went into the developement of Pearl III; there are Pearls I and II, but III combines the features of both and adds considerably to them; III is neccessary if the applications desired are at all extensive. The required tools are CP/M, CBASIC2, lots of memory, the 80X24 mod for Sol, and some basic programming skill; the more one desires from the programs developed, the more is required in knowledge of Basic and programming skills, which is another way to say that Pearl will not do what one cannot do unaided; it just makes it a heck of a lot easier and quicker. Pearl provides the framework, the bulk of the algorithms, entry points for the insertion of data and Cbasic code as needed to create the unique program desired, and eliminates most of the tedious code-keying. It must be said that Pearl is not something one masters at the first sitting; like most DBMS and developemental schemes one must almost learn a new language. It is, however, all quite logical, and once you become aquainted with the idiosyncracies, a joy to use.

Pearl consists of about 50 programs and sub-programs on 5 disks. All this is in source-code and seems to be very complex; the user need not be concerned however. The creation of a desired application is really a matter of following a sequence of steps, all of which are well planned and completely explained in detail in the better than average documentation. The original manual was a bit obscure but Rev.2 includes a much clearer one with good examples and a comprehensive index. There is a rather lengthy example-program with which one gets the feel of Pearl by actually creating a well planned program by copying the steps as indicated.

If one would like to develope his own set of programs for a small-business, and likes to be innovative, then Pearl III is a lot of software for the money. But if one does not like to program and is willing to settle for software that does a lot more than he wants or does not quite do what he wants then some of the off the shelf stuff is the best way to go. Pearl is not available in Helios format (so far as I could find; Lifeboat quit selling it) so I bought it in NorthStar and transferred to Helios CP/M by putting both controllers in Sol at the same time. If there is one big objection to Pearl it is that programs developed with it run painfully slow; that is not news to anyone who has worked with Cbasic I'm sure. I guess we just cant expect to have everything.

There isn't a great deal more that I can say without writing a lot more than most would care to read. We created a very extensive inventory control program with which we keep track of

CON'T ON PAGE 19

CON'T FROM PAGE 18

a lot of merchandise. The data-base has grown to more than a million bytes and we use five different report formats; so you see that Pearl is only limited by the users desire and ability. One can buy manuals for off the shelf software to use as a guide in creating modified versions for ones unique needs. I will be glad to answer any specific questions if someone wants to write or call.

Sincerely,

Earl Dunham

941 N. Russell
La Habra, CA 90631
(213) 697-7238

Jan. 24, 1983

.ASSISTANCE NEEDED IN DEVELOPING SOFTWARE FOR BLIND/D.RUDER

PROTEUS
1690 WOODSIDE ROAD
REDWOOD CITY CA.
94061

To Whom It Concerns;

I am presently developing a microprocessor product for which I am seeking assistance in software development(SD). By my estimate, completion of the prototype requires a 2 man-month SD effort and completion of first release requires an additional 5 man- month effort.

The desired experience and expected program development is Z80 assembly language for real time process control.

The product is an aid for the vision-handicapped, i.e., both those with only partial sight and those totally blind.

My resume is enclosed.

If you are interested and qualified in participating in this software development, please phone me at 415-321-2744 or mail your response to 1 Fenwood Drive, Atherton CA., 94025.

Thank you,

Sincerely

Don Ruder

Don Ruder

.SUPERFILE/Paul Kittle

**** NAILUTCHIJ SIBERIANS ****
Paul and Bev Kittle
Box 1285
Loma Linda, CA. 92354

PROTEUS
1690 Woodside Road. Suite 219
Redwood City, CA. 94061-3483

Dear Stan:

Just got the newest Proteus and am responding to your request for reader input. Couple of things, both info and request for info. First, I am currently selling a program called SuperFile by FYI., Inc. and want to say it works very nice on Sol.(I've got an old set of Micropolis Mod.I drives that aren't supported anymore by Micropolis). The Superfile program is primarily an information retrieval system for data input under wordstar, or any other word-processing program. Once the program has been entered under the word-processing program, including the two or three required flags needed by sf to operate, sf will generate an information dictionary, file and database. Those in small businesses might want to try it on their customer records, special client data, or anything that one might use a conventional (manual) file system for. It all works on keywording, and will allow up to 250 key words per entry. This means if a person is using their system, as I do, for academic work, e.g. dissertation writing, the operator call pull up the data using the Boolean "and", "or", and "not" statements to broaden or narrow the search. In addition, I am currently attempting to write a sample data file for search strategy and techniques that can be helpful in getting used to commercial data bases such as Dialog, National Library of Medicine and ERIC (which is an educational Data base). My interest in Superfile, beside the dissertation application, comes in my Library orientation (I am getting my degree in Library and information management at U.S.C.). Anyway, if anyone is interested in such a program, especially since it only costs $195.00, and I am willing to discount that slightly if I get enough orders. Although it is a somewhat easy system to use, it does take some initial getting used to.

My request for information is in regards to an IBM compatable 8" drive system that will work with Sol. If Helios can do it, great, but I'll need to find someone that would be willing to work with me on the price and a payment schedule.

I guess that's enough for now. I want to upgrade Sol to the 80 x 24 screen, but again, same problem as with the drive system-payment schedule, or lay-away, or something. Libraries aren't the highest paying jobs right now, so my hopes on writing "bigger and better" software is kind of on hold.

A footnote on the Smith-Corona TP-1. I have had a minor problem that would have been major if I weren't an old maintenance man. The problem arrose when I used my TP-1 for really long (20-30 pages at a time) print-outs. The TP-1 started adding a line space where one has not been called for. The solution lay in tightening up the line feed screws on the side. Apparently after 6-8 weeks of daily use with long print-outs these adjustments loosen up and the printer starts adding extra lines. Tightening up the screws with a little "Lok-tite" seems to fix the problem. Still no word on when the tractor feed will be available. (and I see it's now $590.00)

Hope to hear from someone about the drive system and (hopefully!) the 80 x 24. I admit I'm not asking for much. (ha!)

Thanks for the good stuff in Proteus! Paul Kittle

.WHAT MONITOR FOR THE McVIDEO/Michael McKelvey

McKELVEY
& SOL

computer services
to support
the architectural profession

January 29, 1983

Stan Sokolow
Proteus
1690 Woodside Road, Suite 219
Redwood City, CA 94061

Dear Stan:

While I haven't acted on it yet, I remain quite interested in the McVideo (I like all these "McNames", having one myself) 24x80 video upgrade.

I've started looking at video monitors since a new one would be needed to display more characters across. One question I've had is how much bandwidth is needed. The January 1983 Byte had some answers in the "Ask BYTE" column.

I learned that the bandwidth needed depends on how many dots across your character set uses. Characters with a 5x7 dot matrix with 1 blank dot between use a total of 480 (6x80) dots across for an 80 character display. The Sol uses a 7x9 dot matrix with 2 blank dots between for a total of 720 (9x80) dots across an 80 character display. I've assumed the McVideo mod. keeps the same 2 dot separation between characters as a standard Sol.

This means that a monitor advertised as providing an 80x24 display may only have sufficient bandwith to work well with inferior machines using 5x7 characters. Such monitors may not have enough bandwidth for the McVideo mod. even though they claim an 80x24 display and do indeed provide that in some cases.

Fortunately, Steve Carcia provides a way to estimate bandwidth needed, although his math seems a little convoluted to me. If you want all the dots clearly displayed with no smearing, the bandwidth needed equals the number of dots across divided by the active trace time (which Steve says is usually about 42 micro-seconds).

So, based on all of this, the McVideo mod. needs:

80x9/42 = 17.1 MHz bandwidth minimum

Those inferior machines would only need:

80x6/42 = 11.4 MHz

A machine with 7x9 characters with one dot separation would need:

80x8/42 = 15.2 MHz

A 64 character Sol would need:

64x9/42 = 13.7 MHz

It is interesting that a standard Sol is more demanding than many other 80 character machines. The modified Panasonic TV monitor furnished by ProcessorTech doesn't have a 13.7 MHz bandwidth which is why the horizontal dots smear together. It does have the advantage, though, that you can watch Hill Street Blues on it. If you don't mind this kind of smearing, you could probably use bandwidths less than those calculated above.

One difficulty in shopping for monitors in magazine ads is that bandwidth often is not stated. Also, different ads claim different bandwidths for what seems to be the same monitor. I've seem this happen with the NEC JB-1201 monitor. Another monitor I'm interested in is the USI Pi-3. It has a 20 MHz bandwidth and amber colored phosphors which seem to be the coming thing. The amber does look a little strange when you are accustomed to white or green. An advantage to amber is that there is greater contrast between amber and black than between white and black or green and black. The Pi-3 also has the advantage of having the same kind of video connector on the back as the monitor cable we already have uses. Many other monitors use a different kind connector and you would need a new (special) cable.

I've been thinking about other aspects of the McVideo conversion too. Here, I have more questions than answers. Originally, I had the impression that McVideo provided for switching Solos in and out to get the full 64k of RAM. After seeing Fr. McGahee's recent article on his Super Phantom and talking with him, it seems that McVideo doesn't exactly do this. I still have some confusion on this.

If we have the capability to switch Solos in and out, there is no longer any reason to move it to F000 for the 80 character display. There would be advantages to having a 64 character Solos at C000 for PTDOS and old software and an 80 character Solos, also at C000 for both modified PTDOS and CP/M or P-System. Solos could be switched out for non-PTDOS use.

The McDPM Dual Personality Module is nice, but with the above implementation, its memory location switching wouldn't be needed. Could a 2708 personality module be easily modified to switch between two 2716's? I know an earlier Proteus News had a contribution about modifying the 2708 module to use 2716's. Unfortunately, it wasn't clear to us who are ignorant but can follow clear instructions.

Could 2716 switching be accomplished using the the two extra FCh port controlled flip-flops provided with the McVideo modification?

I ejoyed reading about the UCSD P-System. I wasn't aware of some of the new additions to the system. One thing that always bothered me about an otherwise terrific sounding system was UCSD Pascal's lack of numerical precision. I just received some information from Softech and I see that they have dealt with that by switching to four-word real numbers. I've just about given up on getting real numbers for SLAC Pascal.

I've been reading the items about possible Sol backplane problems. If the difficulties are with the connectors, why do we need new printed circuit boards? Couldn't the connectors be replaced on the old board? I'd be happy (perhaps others too) to get in line to buy a new backplane board for insurance purposes if it was more clear why I might need one.

I hope it is not too late to get my Santa Claus list in. I wish someone would show me how to switch between several parallel devices with my Sol. I have the impression that this is

CON'T ON PAGE 21

CON'T FROM PAGE 20

possible with a standard, unadulterated Sol using things with nasty names like PUS. This would allow me to keep my SolPrinter 2 hooked up to my parallel port and add another faster printer, like an Epson, to the same port while keeping my modem on the serial port. Parallel port Epsons are cheaper too.

Cordially,

Mike

Michael A. McKelvey

[Ed. note: The McVideo 64/80 column video upgrade for Sol does use the very same character generator chip that is in your Sol now. You're right that this high dot density requires a good quality monitor. As you mentioned, you can accept a little smearing, and in fact it helps blend the dots into continuous lines.

My recommendation is to try the 80 column mode on a monitor you intend to buy, or compare several, to see if you are satisfied with the image. After all, you are going to be looking at that image for the remainder of the life of your Sol, and nothing is more annoying than a fuzzy, jittery image. I have a Sanyo sitting on the shelf because I couldn't stand its image quality on the standard Sol, but it was a bargain. No bargain if you can't use it. Even if it costs you a little more to get the monitor you want locally, over the life of the device the difference in cost will be insignificant.

I'm sure a 20 MHz monitor will be satisfactory, but there is also the question of phosphor persistence. In the 80 column mode, McVideo takes long enough to fill the screen on each refresh frame that some people's eyes can see a slight flicker of intensity. A phosphor which fades out quickly after the electron beam passes will have more flicker tendency than a long-persistence phosphor. The green P4 phosphor has a longer persistence than the white phosphor in standard TV sets, so it helps reduce the chance of flicker. I don't know about the amber phosphor. (Some sets use a tinted glass front instead of colored phosphors, so color is not necessarily an indication of the type of phosphor.) That's another reason to try before you buy.

About the 64K memory space -- Yes, McVideo does allow the Sol's dedicated address space to switch in and out under software control, in both the 16x64 and 24x80 mode. When in the 24x80 mode, the Sol's space is automatically relocated from C000 to F000, in addition.

About modifying the existing 2708 module -- Anything's possible, but not necessarily neat. The DPM has a switch on the back. I would have preferred software control, but Bob Hogg just chose another route. I suppose the extra flip-flops could be used to provide the switching.

About the p-System -- In the latest manual I received, Softech says that they intend to stop supporting two-word (32 bit) reals someday, and just support four-word real numbers. With the present p-System, you have your choice, but the industry trend seems to be for more precision. Also see the news about p-System for Helios in this issue.

About the backplanes -- It is very hard to remove solder from the 100 soldertails per connector without destroying the PCB. I suppose it's possible, but getting new PCB is the best way.

About the diagnostic programs -- They can be run on a Sol with McVideo in the 16x64 mode. ParaSol should work in either mode, but the software assumes a standard 16x64 mode.]

.TRANSFERRING PTDOS FILES TO OTHER COMPUTERS/N.Pulsifier

NATHANIEL PULSIFER & ASSOCIATES
Investment Management & Financial Planning

P O BOX 500
ODD FELLOWS PROFESSIONAL BLDG.
IPSWICH, MASSACHUSETTS 01938

AREA CODE (617) 356-3530

February 9, 1983

Mr. Stan Sokolow
Proteus News
1690 Woodside Road
Redwood City, CA 94061

Dear Stan,

How can PTDOS files be transferred to other computer systems? Could we use a communications program, modems and the phone systems (providing the new set up is S100 bus compatible? How about plugging something into my SOL?

NP/mz

[Editor's reply: Transferring files from PTDOS to another computer can be done easily if the two computers are brought to the same place. You need a software driver for the port that connects the two. The Sol3 driver will send ASCII to the serial port. The parallel port driver in Solos can be used for parallel transfers, but you will need to write a small PTDOS device driver routine to call the appropriate Solos entry point.

The sending Sol with PTDOS running should be given the command

```
*COPY source,driver
```

where source is the filename to be sent and driver is your device driver name, such as Sol3. Before pressing the return key, prepare the receiving computer to accept data from the corresponding port and to write this data onto the file. Press return on the sending Sol, and you should be on your way.

Be sure that the drivers use handshaking, so that the receiver will signal the sender when it is ready to accept the next character. Otherwise, when the receiver writes a block of data onto the disk, you will lose characters sent by the sender.

Although I've never done it, you should be able to use PIP on a CP/M receiving computer to accept data from the port and send it to the new file. Your BIOS would have to contain the driver for that port, of course. The command on the receiving computer would be something like:

```
A>PIP B:filename=dev:
```

where filename is the name you give to the received data and dev is the device name corresponding to your BIOS's driver for that port.]

.MYPEN/Robert Bartels

Dear Stan:-

I am anticipating a renewal notice in the next issue of PROTEUS/NEWS, so have enclosed a check for my 1983 subscription. I have also anticipated a possible cost increase that always goes along with an inflation.

Letters published in the past few issues of the NEWS have expressed concern over the lack of continued support for the text editor PENCIL developed by M. Shrayer. Having long since thought about writing a version of my PENCIL-I in 8080 assembler language, the needs expressed in the letters provided the necessary motivation to do something about it. After many hours of translating and modifying the disaaembled PENCIL -I version SSN, I am now processing this letter with MYPEN. It is a version that differs from the original in many respects. Disk services are obtained by issuing the standard calls to the North Star DOS. MYPEN does not, as PENCIL does, contain the rudiments of the disk operating system and, therefore, is not independent of N* DOS. Shrayer's method of crossing the memory roadblock imposed by the standard location of DOS (2000H - 29FFH) was to incorporate the necessary elements of DOS in PENCIL and thus permit the text buffer to run into and over the space dedicated to DOS. Hence, with the standard configuration, one cannot exit PENCIL and jump directly into DOS but must go via SOLOS to re-boot the disk operating system. Since my N* DOS and PROM are non-standard and in the memory range F000H - FFFFH, there was no need to avoid the conflict. In spite of sevaral added features, MYPEN is therefore smaller than PENCIL. It includes the capability of creating, writing, reading and listing files on any one of three single-density drives. It recognizes a hyphen as an end-of-line terminator. And not the least, by making use of the SOL Keyboard Customizer (available as ITEM 4 in the Catologe of PROTEUS Products) MYPEN will respond intelligently to each of the 15 function keys on the numeric keyboard. It's a joy to hit a single key and have the printer come to life and spew out the text.

I will gladly donate the assembler language source for my version of Shrayer's PENCIL to PROTEUS if you think that there are enough North Star advocates among the membership. There is no problem in adapting the program to the single-density N* system with DOS in its "standard" location, viz., 2000H - 29FFH. A simple change of an ORG statement is all that is necessary to relocate the text buffer and place it, say, above 29FFH. The program and perameter space stays in the range 0000H - 1310H. Also, changes in the EQUates will accomodate any location of SOLOS. Better yet, any member now owning PENCIL who wants a customized version of the object code and the assembler language source should let me know. A price of $20 would cover the the cost for my time, a floppy disk and the postage.

Aside from the above, I would appreciate knowing whether I can purchase from PROTEUS, or elsewhere, the Z18 keyboard encoder ROM on SOL's keyboard PC-board. One with SOL's original coding. As indicated above, I have the modified IC which was supplied with the SOL Keyboard Customizer and Maintenance Kit (PROTEUS Cat. ITEM 4). I have tried calling Barry Watzman several times but got no answer. I assume that he has been supplying PROTEUS with the modified chip (Z18A). The reason for wanting the Z18 with original coding is because mine went bad. Don't understand why but it did, and it was not easy to verify that it was the cause of the keyboard malfunction. It was not until the bug crept in that I tackled the job of unsoldering the bad Z18 and replacing it with a socket mounted Z18A. But knowing that a ROM can go bad leaves me with a feeling of insecurity. I need the assurance of a workable backup. Maybe a spare Z18A would also be a prudent investment. I will call you in about a week for what information you can give me on the availability of the Z18 and Z18A.

There is an interesting sidelight to the development of MYPEN. Before assembling the object code it was necessary to rewrite my assembler. The assembler language source for MYPEN is about 35K in length. MYPEN is itself about 4K in length. Then add to this 5K for my assembler and the space needed for MYPEN's symbol table (there are just over 420 four-character symbols in MYPEN) and the conclusion is that my 52K of RAM is not adequate for the job. The solution was to split the assembler into the EDITOR and the ASSEMBLER. The EDITOR contains the line file manager, editor and the usual utilities found in an assembler. The ASSEMBLER contains the two-pass assembler and is "disk driven". It reads the source generated with the EDITOR from disk in blocks of three disk sectors as needed, placing the blocks alternately in two buffers. Hence, the space occupied by the source statements is only 1.5K (a disk sector contains 0.25K bytes). The size of the programs I can now assemble is virtually limited only by the size of the source code that can be written with the EDITOR in 52K RAM.

Time to get a bigger comr·ter. I am running out of interesting things to do.

Sincerely yours,

Bob

Robert C. F. Bartels

P.O. Box 2240
Ann Arbor, MI 48106

[Ed. note: Proteus has an electronic parts supplier which can copy the Z18 ROMs. We can copy one from our machine to supply you with a good-as-new ROM with the original keyboard encoding. Call Jane at the Proteus office.]

.HELP! NEED TO DUMP DATA THRU MODEM TO OSBORNE/P. Kittle

**** NAILUTCHIJ SIBERIANS ****
Paul and Bev Kittle
Box 1285
Loma Linda, CA. 92354

PROTEUS
1690 Woodside Road. Suite 219
Redwood City, CA. 94061-3483

Dear Folks:

Does anyone out there have access to a modem program that will work with Sol? I'm running Micropolis drives (single density, single side), and can load to disk using Richard Greenlaw's "disktape.com" for cp/m from the old Solos tape recorder. I need a capture program that allows Sol to capture in terminal mode, then write to the micropolis disk drives, and also allows an Osborne (or anything else) to capture from my drives through Sol. None of the vendors I can find will support my particular set-up, and I need to dump some data from my system (specifically, large portions of my dissertation) into an Osborne that belongs to the school. HELP!!!

On the side, I have worked out a method of using WordStar, Mailmerge, and SuperSort to manipulate citations from commercial data bases, so if this type of thing interests anyone, let me know. All copyright negociations need to be dealt with directly with your commercial data-base vendor.

Thanks! Paul (And a big thanks to Father Tom for his help on the Sol phantom conversion!)

.A BUG IN THE SOLON ROUTINE FOR CP/M/Dick Rathbun

December 28, 1982

Dear Stan:

Thank you for the invaluable service that you provide to myself and other SOL owners as editor of Proteus News and Caretaker of the organization. I look forward to each issue- even the "skinny" ones.

My system consists of a SOL and a Morrow 2D controller with 2, 8" drives running under CP/M 2.2.

I recently completed the SUPER PHANTOM (VOL 5 #2 Proteus News) modification to my SOL and thank Fr. Thomas McGahee for the excellent instructions. The modification works Great!

In the process of installing CP/M in the new 56K system, I discovered that it would be necessary to use the stack swap routines that Father Tom thoughtfully provided. I discovered a small bug in the SOLON routine.

Since the idea of the stack swap routine is to relocate the stack before turning SOLOS on and thus turning the stack off, the routine as shown won't work. It turns SOLOS on before attempting to relocate the stack. To fix this, the first four lines of the SOLON routine should be relocated to just before the RET instruction at the end of the routine. With this minor modification, the routine works beautifully.

Also, at Father Tom's suggestion, I burned a new SOLOS ROM incorporating the bootstrap routine - to turn SOLOS off so that CP/M can overlay it - in the space normally used by the SOLOS TErminal routine. This allows a cold boot for CP/M to be done with only two keystrokes.

Sincerely,

Dick

Dick Rathbun
4653 E. Geddes Ct.
Littleton, Co. 80122
(303) 771-0740

.HELP NEEDED TO INTERFACE A JOY-STICK/Keith Bettis

Keith R. Bettis
2176-a Cloverwood
Scott AFB IL 62225

Hi

I got something I want for Xmas:

1. ...a method of using a 'joy stick' in games, and in the future perhaps in a word processor (like Xerox 6010 Star system)
2. ...a version of Pencil which allows for non-printable control characters (my RS line printer viii uses to underline, change fonts, etc.)
3. ...and many K of $.

Seriously, I would like some help in interfacing a joy-stick or 'joy-ball' (would that be the correct term?) into this system. I want it for games (naturally), curser control for graphics and for word processing work (move this to here type stuff)

Thanks for any help you can give me. I will be ordering some equipment from you when my money builds up from Xmas...look out CPM, here I come.

Long live the Sol,

Keith R. Bettis

UNCLASSIFIED ADS

FOR SALE-SOL 3P+S board NEW & complete with manual. $125.00 George Boettcher Jr,PO Box 100,Dingman's Ferry, PA 18328, TEL: 717-686-1200.

FOR SALE

SOL 20, with 32K, E. Basic, Pilot, Focal, Electric Pencil, Trek 80, 8080 Chess, EVS Accounts Recievable, assorted business/investment programs, Game Pac's 1&2, two cassette drives and black box printer. <u>All offers considered</u>.

Alan M. Becke
3333 Moores River Drive #811
Lansing, MI 48910
(517) 372-5647 or (517) 351-1112 8-5am. EST.

FOR SALE: SOL 20 (Rev E) with 16K ram, all manuals ...$650. With Electric Pencil, EDIT, ECBASIC, PILOT, FOCAL, games, all user group publications since beginning ...$720. Tarbell SD Floppy Disk Controller with CP/M 1.4 & Fr McGahee's utilitites ...$250. Northstar 32k 4mhz dram ...$290. Altogether for $1150 ... or make an offer.
E F (Ernie) Roush, 1779 Walnut Ln, Eagan MN 55122
(612) 454-2866

FOR SALE
North Star DQ controller board and two drives (1 MPI B-51, 1 Shugart SA-400). Complete disc system in perfect operating condition. Configured for SOL so will plug in and run without modification. Also includes about 65 flippy discs with lots of software including Pencil, Northstar DOS & Basic 5.2, CP/M 2.2, games and utilities. Asking $825.00. Dick Rathbun, 4653 E. Geddes Ct., Denver, Co. 80122. Call evenings (303) 771-0740.

<u>FOR SALE</u>

SOL-20 (all or part), Interface 3P + S I/O, RS Speechlab (Heurisics), Computalker CT-1, Expandoram 64K (4116-2) Rev. E, 8800 Interface Board, Extesis Ram Board 6448, with 32K, IBEX 16K PROM Memory Board, 2 MITS Ram Boards, (4K), 1702 PTCO 2K Rom, and Heathkit Terminal H9.

For information, prices or offer, write to:

John Sousa
16086 Darcie Lane
Salinas, CA 93908

or call (408) 455-1324 from 4:30 PM to 10:00 PM, only.

Bob Sparks is looking for a reliable repair technician, PO Box 841, Indianapolis, IN 46206

Roger Lewis wants to trade a 64KRA bare board for EPT #3. Home: (916) 895-3020, Work: (916) 891-7300 X7627.

TABLE OF CONTENTS

PROTEUS/NEWS

A news journal for owners and users of Processor Technology Corporation computer equipment. Published by Proteus, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483,USA, telephone (415) 368-2300.

Submit items for publication to Proteus News, Attn: Stan Sokolow, 1690 Woodside Road, Suite 219, Redwood City, California 94061-3483, USA. Please make submissions as camera-ready as possible by using a fresh, black ribbon and typing single-spaced.

FROM:
PROTEUS
1690 WOODSIDE ROAD, SUITE 219
REDWOOD CITY, CALIFORNIA 94061-3483
USA

MAG 30 172900C1 03/29/83

MAGUIRE'STEPHEN A
626 N DODGE BLVD #10
TUCSON AZ 85716

RENEWAL NOTICE

TIME TO RENEW

This is the last issue of the current subscription year, 1982. We always seem to be a little late, but it's better to have renewals come after everyone has recovered from the Christmas holidays anyway.

The new subscription rate for 1983 will be the same as last year's. Although inflation did raise costs a little, we seem to be having thinner issues, so this offset the inflation.

1983 WILL BE $30.00 PER YEAR FOR U.S., CANADA AND MEXICO, AND $38.00 PER YEAR FOR OTHER FOREIGN ADDRESSES.

Let's all try to fatten up the issues by writing more. Thanks for those articles that came in response to my last request.

Stephen Maguire

2430 N. Dodge Blvd.#250
Tuscon, AZ
85716