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 Sol (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 high-quality during update of the video RAM; (3) it allows the Sol's dedicated address space (ROM, 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 display board is capable of. This 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

NSW 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 PerScsi 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 manages data transfers to and from the calendar clock maintains date and time-of-day down to fractions of a second, and it continues to function with its 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 controller chip, which means that it is able to perform with any PerScsi drives. Many controllers on the market now use the NEC chip or equivalent, which is an elegant floppy disk controller but can only step the drive at 1 ms maximum rate. The VFDC 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-sectoring format (IBM 3740 single-density format, IBM system 34 double-density format).

Moreover, the controller and software know all about the PerScsi drive's unique features. Many controllers are totally confused by the fact that the PerScsi 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 PerScsi are non-standard, so most manufacturers do not support the PerScsi directly, requiring cuts and jumpers on the controller and/or the drive. This board is capable of doing this on its own. It 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 PerScsi signals for Seek Complete, Spin Up Motor Enable, and Direct Head Load are all supported by the board.

CONT'N ON PAGE 2
conjunction with the MicroComplex Dual Personality Module. Together, these two boards will provide the old 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. Software that uses the memory-mapped video directly, however, will need modifications to know the new origin (PC0) and size of the display. I have written a disk-accessed PTDDS as 16x40 memory-mapped EDIT. The navigational edits (the ones that "yank" 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 because it will have to be used in the 16x40 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 1-80 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 does about twice as much, so the 8088 board will probably be used only for Sol and may be available around 1984.

The Dual Personality Module, programmed with the new McSolos, is available from Proteus for $95. The McVideo board and the 8088 board are available in sets from Proteus; fully assembled. Not a bad plug-in for those with a Commodore 64. I can't guarantee that it makes the Sol over in so many ways. And I can't promise you will be happy with it. If not, let me know why and we'll try to fix it in condition within 10 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 IBM's PRAM disk drives or to a PRAM "SMART" interface which connects four PRAM 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 digits. That is, the following digits can be read: thousands of seconds, hundreds of seconds, tens of seconds, seconds, minutes, hours, day, month, and year. For example, Wed, day of month, mon, month, year, is 1-1-77. The battery provides approximately one year between battery changes under normal use. When the system is turned on, the clock chip keeps track of the computer; when it is 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 files when they are changed, so that you can be certain of which file is the latest. Programmers will find this handy for keeping proper code revisions in order during frequent debugging sessions.

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 see. You also need to be aware of the attempt to break the security of the system so the culprit can be found. This is a key technique in detecting and preventing attempts in financial systems.

The clock would also allow high-security systems to have an automatic lock-out function based on 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 so. Using Wordwizard might be like that, making this exhaust 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 flaky 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 PO - D7, so the board can be used in systems 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 controller, fully assembled. Not a bad plug-in 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 in 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 multiplexor feature of the new VFDC board. They are able to pass data from the Helios format to the single-density IBM format with 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 learning how to 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 DISKAPE utilities move CP/M files to and from Sol's cassette tape. Our tape librarian Lewis Moses can provide you with these utilities.

The board does provide an automatic bootload ROM, so you must have a bootload routine (about 20 to 30 bytes) in ROM.
somewhere. A modified version of Solos in 2708 or 2716 EPROMs, both having the Helios bootloader and the VFDC controller’s bootloader 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 diagnostic utility, 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 restocked version, known as Solos 2.2 for VFDC in its standard Solos is $170. Included with the CP/M are a diskcheck command (similar to the MDOCS DCECK), file compare, and disk recovery programs.

The modified bootloader—Solos in EPROM (2708 or 2716) is $20. If you have one of the old personality modules that does not have the 2708, 2716, or 9216 ROM Solos, you will need to get another personality module. We recommend the dual personality module made by Microcom, for $95. This has been described in previous issues of Proteus News. It is available from Proteus also. When ordering, specify that you want the bootloader for Helios and VFDC. This personality module has the advantage that it is compatible with the Microcom 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 will allow the upgraded Sol to work with 64K BIOS software, 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 Prism hard drive 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 $500. Also available are 10 or 70 megabyte versions, let me know if you are interested.

65K RAM Memory

by Jim Spann

Reprinted from "Microsystems" magazine.

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 VCM 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 work properly.

A minor wiring change and the addition of two integrators will give the Sol Terminal Computer capability of the Sol Terminal Computer. This feat is accomplished by the use of a 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 1. Build the custom memory management control circuit as shown in figure 3 on a small vector board. This board 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 pull-up pins of the S-100 bus to the Sol’s internal bus. Set this board to one side. All the parts required are 74LS74 (flip-flop) and 74076 (open collector inverter).

The required circuit changes can be made without cutting any printed circuit board traces. The technique of wiring 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 (Fig. 2). The two-byte memory management control circuit can be assembled on a small vector board and mounted inside the Sol using the keyhole shown in figure 2.

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 powered on, the memory control flip-flop is cleared via the Power On Clear signal (POC). This signal is also generated when a reset is performed (holding both the upper case and lower keys down). The Sol will operate normally with the Sol’s display RAM/ROM memory block enabled.

The memory control flip-flop resets the system at boot time (read/write to the C000-CFFF hex memory block). This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system). This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system. This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system.

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 (1) 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 step:

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 inside the C000-CFFF address area.

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 powered on, the memory control flip-flop is cleared via the Power On Clear signal (POC). This signal is also generated when a reset is performed (holding both the upper case and lower keys down). The Sol will operate normally with the Sol’s display RAM/ROM memory block enabled.

The memory control flip-flop resets the system at boot time (read/write to the C000-CFFF hex memory block). This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system). This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system. This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system.

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 (1) 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 step:

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 inside the C000-CFFF address area.

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 powered on, the memory control flip-flop is cleared via the Power On Clear signal (POC). This signal is also generated when a reset is performed (holding both the upper case and lower keys down). The Sol will operate normally with the Sol’s display RAM/ROM memory block enabled.

The memory control flip-flop resets the system at boot time (read/write to the C000-CFFF hex memory block). This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system). This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system. This block can be used to display RAM/ROM or a 4K RAM (can be partitioned) in a larger memory system.

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 (1) 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 step:

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 inside the C000-CFFF address area.
The following is an example of a C80 family interface using the memory management modification.

```assembly
// Procedure to set up the interface
set_interface:
    ; Set the memory map
    mov cl, 0
    mov ah, 0
    int 21h
    ; Set the interface registers
    mov al, 0
    mov bl, 0
    mov ch, 0
    mov cl, 0
    mov dl, 0
    mov sh, 0
    mov dh, 0
    mov bh, 0
    mov ah, 0
    int 21h

; Procedure to send data
send_data:
    ; Set the interface registers
    mov al, 0
    mov bl, 0
    mov ch, 0
    mov cl, 0
    mov dl, 0
    mov sh, 0
    mov dh, 0
    mov bh, 0
    mov ah, 0
    int 21h
    ; Send the data
    mov al, data
    out port, al

; Procedure to receive data
receive_data:
    ; Set the interface registers
    mov al, 0
    mov bl, 0
    mov ch, 0
    mov cl, 0
    mov dl, 0
    mov sh, 0
    mov dh, 0
    mov bh, 0
    mov ah, 0
    int 21h
    ; Receive the data
    in al, port
```

HARDWARE INTERRUPTS FOR THE SOL

by Stan Sokolow

Leon Winter in Bukidnon, Philippines, has written asking about using interrupts to drive his Mytype 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/Heilos. 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 a 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.

One problem 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 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
It may be of interest to PROTEUS NEWS subscribers that PerSci 277 Disk Drives are being sold at very reasonable cost by Gantel Corp. (Hawward, CA). Actually, the drives are being sold by Gantel to the lack of the P19 & P20 (while Protect 1 & 0) on the top row of connectors of the Data and Interface PCB.

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

--- PerSci/Gantel 277 ----> PerSci/Processor Technology 270 (DAI PCB 200263-012) (DAI PCB 200263-007)

1. Change R8B & R94 to 4.75 kohms (1/4 W 5%).
2. Change C40 & C41 to 1 microfarad (Tantalum, 35V).
3. (All of the above components are found just to the right, and left of U10 and establish the index & sector pulse widths from it).
4. Change U11 (select Module DIP) to one described in the Helios II Disk Memory System Manual, Section 6, Fig 15.** or, if only one drive is in the system simply Delete U11.
5. Jumper Delete Add 270 Function
   a) W-D W-Y Sector Ser One Shot Tris 0
   b) AB-AC AB-AA Ser. Sector Pulse Outut
   c) AB-AE AB-AF Ser. Sector One Shot Tris 1
   d) AH-AJ AH-AK Ser. Index Pulse Output
   e) AW-AV AW-AU 45 to U11.14
   f) BH-U11,10 BH-BJ Ties Ser. Sector to Data Ser.
   g) U325-U32,1

--- Third --- the front bezel diskette eject switches are not active as Gantel 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 relays)

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.

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 Dual Disk Drive (as is):**

- Richard Axts: Purchasings
- Gantel Corp.
- Hawward, CA

**Helios Power Supply PCB (bare board):**

- David Tanconie
- Silicon Technology
- Box 70421
- Sunnyvale, CA 94086

**Helios (and Sol.) Transformers:**

- Deslin' Electronics
- 735 Loma Verde
- Palo Alto, CA 94303

--- March 25, 1982 ---

by David Reis
IMSAT Cabinet for one Per-Sci Disk Drive . . . . . . . . . $50.00
W. Consumer Supply Inc.
1771 San Jose CA. 95112
(408) 295-711
(The Helios Power Supply PCB & Transformer fit in this
box conveniently. The Helios Indicator PCB is
also available to this cabinet by drilling some holes in
the front panel for adjusting the DPEDS and attaching a
photo-reproduction of the indicator panel depicted on
slide 4-4 of the Helios User's Manual for front panel dress).

Assorted Helios Components and Systems

Charles Terry
Special Pass
4211 Bellaire Blvd.
Houston TX. 77025
(713) 660-3136 Home
(713) 666-3515 Work

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 cannot be guaranteed.

David Reis
1843 Vassar Avenue
Mountain View, CA 94043

Good News!

TAD Enterprises has just completed a new software product
for systems that utilize Processor Technology SOLOS or OUTER
and CP/M. The program converts Processor Technology
Extended

Cassette BASIC to a DISK Basic that runs under the Digital
Research CP/M 1.2 Disk operating system. Some significant items
are listed below:

- Easy to install - No knowledge of assembly language needed
- All tape functions have been converted to DISK
- Tape-disk tape transfer utility program is included
- Program will relocate to any 2K or larger CP/M system
- System is supplied on a 1200 baud COMMSET cassette
- User must supply own copy of BASIC Rev. A
- Program uses SOLOS/OUTER 1.0 for console communications
- Program overlays the CCP for maximum memory utilization
- Features include a program TRACE command, User definable filetype
- User printer driver, SOLO 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 easy as loading
and running the utility program included on the supplied cassette.
(Patches are also included for BASIC conversion.)

The basis on CP/M system requires 2K 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.
(depending on how big your BIOS is)

Price of $49.95 includes cassette and user 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 527
Hazelcrest, IL 60439

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

(Program will also work with CP/M 2.0)

(Ed. note: Funny you should ask about the EKBASIC 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 is adapted to CP/M and will be available in
the future. --Stan.)
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 no memory damage. About the time I was coming around again I saw Mr. Allen's article in KiloHaus where he mentioned moving his sense switches to the front panel. Sounded like a good idea at the time, but, after a few hours of spastic desoladering I had fairly well ruined the Sol's main circuit board. This was about the same 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 and U54-10 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, Microcomputers. "Extending the Altair Bus" so I got myself a big blue Vector U1 box, a Sunny International power supply and an 88 slot motherboard, on sale somewhere. The board had the same Godbout on it and seems to have been made for the Vector box. Then I soldered four, five foot 50 conductor cables to two cut-down 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 box constractor. The whole thing turned out pretty well. I was lucky with my handsaw 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-in on-off switch, a five inch speaker with a volume control, a resonator, and a surplus sol pet, although my two Harro 32k boards aren't cooking each other in there any more with all that room. I call the thing Gini's Pole-Lution, to all 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 are saving something and such. I note that some people have mentioned that the GRT Microsoft BASIC tape I bought 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 leads aren't bent before inputting from the tape, so you need a speaker to hear it. Can't see any advantage to the BASIC troubleshooting. 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 Gini

---

The following is a program for the ADS Noisemaker which will give you a 38 key piano. I've got my Noisemaker at home. I've got my Noisemaker attached at 1/0 DF048, the keys used are "g" through "r" and "v" through "z" above "g", the note G is middle C and "d" below "g", "b" below "r", "v" and "z" are the sharps. I calculated the frequency for the notes assuming that the clock frequency for the Sol is 2.048MHz, 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...
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 DISRT) 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.

---

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 ------ (210) 772-1106.
Dear Stan,

I have been running a 56K CP/M system with Solos at F800H for a couple of years now and have often thought it would be nice to squeeze a little more memory into my system. I run Morrow's D20 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... Now 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 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.
2. Re-insert U-34 in socket.
3. Connect a wire from pin 6 of U-22 to pin 1 of the 74LS08.
4. Connect a wire from pin 6 of 74LS74 to pin 2 of the 74LS08.
5. 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 P-P being reset. This allows the signal from 6 and 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 MULTIWITTER DRIVER
by Wayne Wilson

In the May-August issue you printed my Custom I/O printer driver for an ASL2 Multewriter 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 print individual linefeeds. 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:

---

FIXING SOL TO "VANISH"
by Wayne Wilson

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

8

---

CON'T ON PAGE 10
REVIEWS 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 80x24 display was like having a new computer, I personally feel that the 80x24 display is that extra something missing from the SOL.

The board is 8x80, well layed out and nicely silk screened. It uses the Intel 8275 CRT controller with an on-board 8855 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 80,000 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 7x10 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 7x10 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 in a 9x10 matrix allowing high-resolution character generation. 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
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 Solas and to my knowledge they feel as I do, that it has added new life to the old beast.

Yours truly,

Wayne Wilson

---

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 40 years. In a desperate search for a replacement I bought a replacement (same across a company called Memory Merchant), selling a 16K static for $175 including shipping. It took just over a week to get the board, and the sale 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 2000H for my North Star system, and could supply instructions for relocating to any other page boundary (e.g. 100H for DP/W). I also have a set of disk I/O routines that use the NM 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, PTODS, 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,

John Ousadar

---

F. O. Box 1851
Homewood, IL 60430-0451

---

CONTINUED....
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-120 monitor which is also sold by Micro Complex. I agree with all the favorable comments 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 screen processing. Its split screen capabilities are lost on me with the old VIM. 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 2-80 upgrade Bob Hogg is working on. It will allow us to run 2-80 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

Ringing the Bell on a Serial Device Before Printing
by David L. Dalva II

I have recently acquired a Micro Complex 80 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 Personality module. Solder in a small jumper connection. Plug in the 20 by 80 module. Plug in the new Personality module. Test the SOL before assembling it.

The board works well. In CP/M you can run a 56k system on an 24 by 80 SOL screen, which allows almost all of the CP/M software to run without excessive changes.

Micro Complex has also made use of the LOAD key on the SOL to boot the system at either CB08H or CB00H depending on the setting of the switch on the Personality module which also enables the screen for the correct size, 16 by 64 at CB08H or 24 by 80 at CB00H for CP/M. You have also added some new commands to the Personality module including a memory test, a ASCII dump, a fill command, enter ASCII 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 list 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 all ports A and B.

I have been using PT's Business Basic in the D000H version and have found that it works very well in a business environment. The Common of command allows the passing of variables from one program to another. You are able to lock out the MODE SELECT key and 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.

Sincerely,

David L. Dalva II

HELP NEEDED INTERFACE N&0DS 5.2D TO SOL
by Robert R. Walling

Dear Mr. Sokolow:

I recently purchased North Star's new DOS 5.2D/BASIC and thought I would be able to use the NorthStar InOut/DOUT routine for SOL Computer written by Joe Masire in volume 2, number 3 issue of the newsletter. I found that I was not able to interface these routines successfully with the new DOS 5.2D. I thought perhaps Mr. Masire 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
9208 Edmonton Rd.
Arlington, Va.
22207

X4S = Name of report to be printed

18 REM. : Test printer before allowing printing
30 REM. : Program name = PRINTS
40 REM. : Created = 28 Dec 1980
50 REM. : Version = 1.01
60 REM. : Last modified = 4 Jan 1981
70 REM. : Checksum = EBD1
80 REM. : Ref. = none
90 REM.
91 REM.
92 REM.
93 REM.
94 REM.
95 REM.
96 REM.
97 REM.
98 REM.
99 REM.
100 PRINT "SET CP=0: CURSOR 1,0; PRINT "; CURSOR 1,18"
105 PRINT "IF Diablos sounded then printer is ready to proceed"
110 OUT 249,7: REM.
115 PRINT "Sends bell on Serial device"
120 CURSOR 2,0: PRINT "; SET CP=0: PRINT "; CURSOR 2,9: PRINT "; SET CP=0: PRINT "; CURSOR 2,17: PRINT "PRINT NOW PRINTING :\n"X4S"
140 REM.
150 REM.
160 REM.
170 REM.
175 REM. : If you have installed the Micro Complex 80 by 24
180 REM.
190 REM.
195 REM.
200 REM.
205 REM.
210 REM.
215 REM.
220 REM.
225 REM.
230 REM. : then the command "OUT 252,7" will cause the SOL's
235 REM.
240 REM.
245 REM.
250 REM. : bell to sound.
I have promised for some time to send some of the information that I have acquired on the Sol Systems along with the comments and notes in the latest issue (Vol. 4, No. 4) that I received today. I have been working on this project for over a year and in the latest Sol Systems, SD Systems was a little help in providing the correct jumper configuration, but once this configuration was established the boards worked well. The only trouble I had was related to the ICs. The boards were built from kits. I am using the following modifications (referring to the pin numbers marked on the Endiagram I Board):

- **Connect Jumper:**
  - E32 to E33
  - E19 to E26
  - E7 to E8
  - E14 to E11
  - E10 to E17
  - E22 to E24
  - E26 to collector of 01
  - U10 Pin #2 to U1 Pin #14

- **Cut Jumper:**
  - E31 to E32
  - E21 to E26
  - E11 to E12
  - E23 to E20
  - E26 to collector of 01
  - U10 Pin #2 to U1 Pin #14
  - U4 Pin #3 to U4 Pin #14

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 in the development of this system I suggested to my students who were working with the Sol that it might probably be handled by a series of CMOS gates and he came up with a circuit using the 74C96 IC. The 74C96 CMOS non-inverting buffers were selected because each gate has an approximate propagation delay of 18 nanoseconds when operated on a +5 volt supply. The delays of 18ns, 15ns, 25ns and 35ns can then easily be achieved by connecting two, three, four, and seven of the 74C96 buffers in series. The delay was removed from the PC board and a socket installed in place of it. Then the two ICs were used one on top of the other (piggyback fashion) by bending and soldering the appropriate 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 Office Systems of Palo Alto for some time. The Selectric unit they sell was originally installed in a CIE terminal. For this function CIE added a PC board to the bottom of the bare Selectric 725 mechanism and installed it in their case. Pacific Office Systems bought the used unit, added their own PC board (still using the CIE addition) to provide an interface to a standard parallel port (I believe they also have a serial port) 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 Sol13 driver on one of the output devices I had at the time. Sol13 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 Sol13 program, and it might be further on:

- IN F8 to IN FA
- AMI F5 to AMI 4
- CPI E0 to CPI 0

After using this driver for a while I found I was getting a more sophisticated driver. (By this time I had modified the 72S mechanism with a 12 tooth index ratchet gear to a 715, a single pass film ribbon, with a 54 tooth index ratchet gear and adjusted the index mechanism so that it would perform line indexing under computer control; thus allowing the user to enter any subscript on the word processor documents.) So I further modified the Sol13 driver to support underlining (by using the back space) and to pause (by typing a non-printing character, SCX or SCY while the tape 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 had time to finish the final. So I will send out what I have written and say that I am operating on 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 inquiries on the development of either of these systems.

For the next several months I will be in Jeddah, Saudi Arabia and can be reached at the following address:

Dr. D.D. Howe
Department of Applied Sciences
College of Engineering
King Abdullah 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.D. Howe

**SOL-20 BACKPLANE PROBLEMS?**

by Bob Marsh

Are you having problems with interlaced system operation that goes away if you turn on the S-100 cards or unplugging the board? This seems to be a common complaint of SOL-20 users, caused by the low spring tension of the T1 connector used by PCT. (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 Ballys 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 printer, in an ad for the printer. Initially, I was certain that this was the printer for me. The price is $1695 for the kit, and $1795 for the assembled unit, and I got the kit for $1,099.20. On 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 most easily damaged components, so the only circuit board that needs to be assembled is the controller, the other three boards are assembled (and tested!), thereby increasing your chances of having a working unit the first time. It took me about 10 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 three major assemblies, not counting the cabinet. These are the printer assembly, the Paper Feed assembly, and the electronics assembly.

The printer 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, a buffer, and the controller. These are all air-cooled by a fan. There are two boards used, a Head Driver, a Motor Driver, and a Logic Board. The printer is controlled by a Z80 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 straightforward, 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. I re-assembled the printer, I noticed that one of the ribbon cables did not appear to be seated in its connector, which was a big mistake. 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. I returned the paper to the store and factory parts support with the latter case, the parts were on 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: descendants, bi-directional printing, underlining capability, select and adjustable print pitch. The H/Z-25 has all that and more. Add double-width, 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, and 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 CT8 and RTS lines, and DATA IN and DATA OUT lines must be cross-coupled.) I am using the same printer driver that I used with the H-14. Many of the printer's 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 imbued 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 (POM, 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 reliability, but for now, I am totally satisfied with my H/Z-25.

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

In my opinion, I think it's 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.

\[ \text{MICROBYTE IN SAN JOSE HAS SOL PARTS FOR SALE} \]

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. 5cl-20 running no TAB boards - 750.00
2. 5cl-20 , power supply had diodes 750.00
3. IECA FAX board parts missing - 50.00
4. Micropolis single drive with controller
   model plus cable, diskette and P.B. - 400.00
5. Format controller FAX board $1450.00
   parts 750.00
6. two PUTS two boards - 30.00 each.
7. GEM with Cutter FON ACT - 20.00
8. IAC 5K TAB (2102c) with FAX works - 20.00
9. Teltronix 7502 dual trace storage scope
   $50.00 Ekh with probe - 500.00

Call us or write us to order. Please enclose check or M.O.
April 1, 1982

(CATLGUE OF PROTEUS PRODUCTS AND SERVICES)

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

SOL & J-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: McPDI. $95.00
Replaces Solos personality module and has twoEPROMs (2716). Switch on back of module (accessible 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 as below.

The new McSolos is an altered version of Solos which replaces the tape I/O commands with other commands, such as Tipto Ascle, 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 its first; explained in installation guide.)

Configuration B = Empty socket C000 and board jumped 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 correctly 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.
3. Tarbell single density controller.
4. Versatile Disk Controller (Proteus item M5). (Controllers that boottload by executing an address that ends in double zero, such as F000, will boottload 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 this module EPROM 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 McFloppy (fixed disk operating system included) is not ready for 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 218 keyboard encoder ROM on SOL's keyboard control board, plus 16 re-legendable keytops and spare parts for keyboard. Full instructions included.

The new ROM encoder sets the high order bit (bit 7) 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 on the keypad 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 in the set. It should then substitute other characters for these special keys.

A 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 page insertion, deletion, previous-page, next-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 key descriptors in Byte magazine, March 1982 page 477, available from FMN Software, 111 Sawmill Gulch Road, Pebble Beach, CA 93953. (408) 373-5103. This routine interprets 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 hard disk / tape interface, and a 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 Perseus 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 adapt 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 the 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, 75, 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 backup 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 utilities, diagnostics, etc. Text editor with special command for reading time and date into the edited text is available. See software section of catalog.

Instructions: Download the Helios boards from the Sol. Unplug the disk drive cable from the Helios controller board. Plug the cable onto the Versatile Disk Controller, plug controller into one Sol disk controller 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 MCDM or Bootload ROM to provide bootloading. MCDM (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 $25.00
Capacitor and fuses, with complete instructions for fixing a design fault in Helios power supply that will in time burn out and cause serious damage to the disk drive. See Proteus News vol. 4 number 3/4 for full explanation.

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: PCT Memory Boards... $45.00
Volume 7: PCT 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).

DOCUMENTATION

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 system with VDC... $15.00

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. On Helios disk in PTDOS format. $50.00
ALS-8 manual and users notes in EFT Vol. 5. 

Item P15: Source code for PTDOS, the Processor Tech disk operating system, including most command files. License for personal use. On Helios disks (3) $100.00
90.00
Item P16: Helios diagnostic programs source disk.
On Helios disk
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), PT DOS \textit{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
Manuscripts 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
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.

NEW!

Item P20: Nevada PILOT.
A new 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,
statistical history taking, programmed instruction and testing.
Available in most disk formats. Requires 32K CP/M disk system.
Specify which type of disk you have.

NEW!

Item P21: PTC Cassette PILOT on executable cassette.
Manuscript in EPT Vol. 4.

Item P22:
PTC Extended Cassette BASIC executable cassette.

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

Item P24: SOLT
SOl test program for diagnosing hardware problems
On executable cassette
Manuscript 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 PT DOS 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.

NEW!

Item P26: VDC Edit.
A CP/M-compatible editor resembling the PT DOS
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-sector diskette

SUBSCRIPTIONS AND BACK ISSUES

Item P42: Proteus News back issue set, 1979............$15.00
Item P43: Proteus News back issue set, 1980............$18.00
Item P44: Proteus News back issue set, 1981............$24.00
Item P45: Proteus News, current subscription, 1982..............$30.00
(NOTE: Proteus News is a periodic and thus exempt from
California sales tax. Prices include postage.)

SERVICES

Item P51: Disk updates
We will update your PT DOS 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 P52: Membership List (Partial)
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

Page

Foreword to the Encyclopedia

Preface to Volume 1...

Chapter 1. Historical Scrapbook of Processor Technology...

"VMD-1" (Jan 1976) ...................................................... 2
"BYTE''s Audio Cassette Standards Symposium (Feb 1976) ........ 4
"AKRA" (March 1976) .................................................... 6
"Altair" (April 1976) ....................................................... 8
"8080 Expansion" (June 1976) ........................................... 12
"8080 Compatible" (Sept 1976) ......................................... 14
"Sol-20" (Nov 1976) ....................................................... 16
"Sol" (Dec 1976) .......................................................... 18
"Intelligent-Terminal Designers Opt for 8080-
-Compatible Circuitry," by Lee Felsenstein and
Robert Marsh (Feb 1977) ............................................. 22
Personal Computer Retailer, Vol. 1 No. 1 (March 1977) ............ 23
The Small Computer Catalog, May 1977 ............................ 31
"Sol: The Inside Story," by Lee Felsenstein (July 1977) ......... 53
"The Sol-20 Computer Terminal," by Rudolf Hirschmann
(Aug 1977) .................................................................. 59
Sol System Suggested Retail Price List (Sept 1977) ................. 61
"Why Sol Should be your Small Computer Choice," ............. 63
"Micro 11 Disk Memory System," (Product description) ........... 64
Sol Small Computer Systems (Catalog) ................................ 70
Personal Computer Retailer, May 3, 1978 ............................ 78
"Build an all-in-one home computer," by J. Free (June 1978) .... 84
Sol System Suggested U.S. Retail Price List
(July 1, 1978) .................................................................. 85
Personal Computer Retailer, July 1978 ................................. 87
Technical Training Seminars Diploma .................................. 91
Sol Dealer List, August 1978 ............................................. 94
Personal Computer Retailer, Sept 7, 1978 ............................ 95
Product Description: "WordWizard Electronic Typing
System" .......................................................................... 101
Sol System Suggested U.S. Retail Price List
(Oct 1, 1978) .................................................................. 105
Retailer Memo, October 12, 1978 ....................................... 109
Personal Computer Retailer, November 20, 1978 .................... 113
Personal Computer Retailer, December 18, 1978 .................... 119
Personal Computer Retailer, January 25, 1979 ....................... 125
Processor Technology 90 Day Limited Warranty, 2/24/79 .................. 129
"Processor Technology Goes Under," by Tom Williams
6/25/79 ........................................................................... 136

Chapter 2. Processor Technology ACCESS Newsletter
Vol 1, No 1, February 1977 ................................................. 2
Vol 1, No 2, April 1977 ..................................................... 14
Vol 1, No 3, June 1977 ....................................................... 30
Vol 1, No 4, November 1977 .............................................. 50
Vol 2, No 1, March 1978 ..................................................... 86

Section 1. Exploring the Sol System ................................. 8
Section 2. At the Controls ................................................ 18
Section 3. Introduction to Software ................................... 22
Section 4. Loading Programs from Cassette ......................... 28
Section 5. At the Keyboard (Key Functions) ......................... 35
Section 6. Unpacking and Installation ................................. 45

Section 7. TLC for Your Sol (Light Maintenance) .................... 54
Appendix 1. Products Available from Processor Tech ............ 59
Appendix 2. Cassette Recorder Supplement ......................... 65
Appendix 3. Selecting a Cassette Recorder ......................... 65
Appendix 4. Interconnecting Requirements for Two Recorders .... 65
Appendix 5. Internal Controls (Switch Settings) ....................... 68
Appendix 6. Keyboard Tables ............................................. 76
Appendix 7. Sol Specifications ............................................. 76

ENCyclopedia PROCESSOR TECHNICA

Volume 2

SOL Technical Manual

Contents

Page

Chapter 1. SOL Assembly Manual
Introduction and General Information ................................ 5
Power Supply Assembly and Test ...................................... 13
Sol-PC Assembly and Test ............................................. 37
Personality Module Assembly ........................................ 85
Keyboard Assembly ..................................................... 95
Cabinet and Chassis Assembly ........................................ 99

Chapter 2. Theory of Operation
Introduction .................................................................... 3
Overview .......................................................................... 3
Block Diagram .................................................................. 5
Power Supply Circuit Description ...................................... 8
Sol-PC Circuit Descriptions ............................................ 10
Keyboard .......................................................................... 40
Timing Diagrams .......................................................... 48
Sol-PC Connector Pinouts .............................................. 55

Chapter 3. Maintaining the SOL
Engineering Drawings ..................................................... 62
Rev D SOL-PC including Keyboard ................................... 3
Personality Module PM5204 ............................................. 13
Reference Material .......................................................... 15
Rev E and later SOL, assembly and schematics ..................... 17
SOL-20 Parts Lists .......................................................... 42
Color Codes ...................................................................... 52
9080 Operation Codes & Common ASCII Characters .......... 53
Loading DIP Devices ....................................................... 111
Soldering Tips ............................................................... 54
Installing AUCAT Pins .................................................... 55
SOL-PC's Pinout Diagrams .............................................. 125
Trouble-Shooting the SOL .............................................. 67
Introduction ...................................................................... 68
Handling and testing precautions ...................................... 68
Basic trouble-shooting procedures .................................... 68
Common symptoms indicating failure in SOL-PC ................. 70
SOL diagnostic programs ............................................... 76

Chapter 4. SOL Hardware Modifications
SOL-PC Board Revisions .................................................. 91
Rev D to Rev E ................................................................ 2
Sol Parallel Prosthetic ...................................................... 6
Rev E to Rev F ............................................................... 9
Rev F to Rev G ............................................................... 12
Rev G to Rev H ............................................................... 11
Rev H to Rev R .............................................................. 12

ParaSOL Debugger Manual ..............................................
Sol Change Notices (PTC-Originated)
Change #HSC Rev A: Heat sink compound ............... 21
Change #2: Assembly procedure .......................... 22
Change #3: Sol KBM modifications ........................ 23
Change #6-2 Rev A: Crowbar fix, Flat Washers ......... 24
Change #7: Relay drive current fix ....................... 26
Change #9: Sol-20 transformer fix ....................... 27
Change #10: Serial current loop fix ..................... 28
Change #11: 2708 Personality Module resistor ........... 30
Change #13: Side panel assemblies ...................... 31
Change #14: Sol power supply ground fix ................ 33
Change #15: Sol audio tape I/O schematic ............... 34
Change #16 revised: Vectored Interrupt ................ 37
Field Service Aid #2: Serial half-duplex fix .......... 39

User-Originated Changes
Run/Stop Circuit ........................................... 47
Audible keyclick for Sol ................................. 48
Cromemco Bytesaver modification ....................... 50
Keyboard mod: UpperCase on power-up/reset .......... 51
134.5 baud Rate Generator ............................... 52
Sol Super-Phantom ........................................... 53
65K RAM Memory Modification for the Sol-20 .......... 55

ENCyclopedia Processor Technica
Volume 3
Programming the SOL Computer

Contents

Chapter 1. Intel 8080 Assembly Language Programming ... 1
Chapter 2. Solos/Cuter User's Manual ..................... 1
Chapter 3. Solos Monitor Program Source Listing ........ 1
(Ver 77-03-27)
Chapter 4. Solos Flowchart (Ver 77-03-27) ............... 1
Chapter 5. Consol Source Listing .......................... 1
Chapter 6. Cutter Source Listing ........................... 1
Chapter 7. Sol Interface Programming ..................... 1
Introduction ................................................. 1
Communications Interfacing ................................ 1
Serial Port Interfacing .................................... 1
Parallel Data Interface .................................... 1
Switch Function Definitions ............................... 1
On-Card I/O Port allocations ............................. 1
I/O Connector Pinouts .................................... 1
Video Display Programming .................................. 1
Memory Map .................................................... 1
Reverse-video .............................................. 1
Special characters ........................................ 1
Scroll ......................................................... 1
Window shade ............................................... 1
Snow-free updating ........................................ 1
24K80 modification ........................................ 1
Cassette Interface Programming ......................... 1
Motor control ............................................... 1
Writing cassette data ..................................... 1
Reading cassette data .................................... 1
VAMIR Control Programming .............................. 1
Audio Alarm Programming ................................ 1

ENCyclopedia Processor Technica
Volume 4
Programming Manuals

Contents

Chapter 1. Software #1 ...................................... 1
Chapter 2. BASIC-5 User's Manual ........................ 1
Chapter 3. Extended Cassette BASIC User's Manual .... 1
Chapter 4. Cassette FOCAL User's Manual ............... 1
Chapter 5. Cassette PILOT User's Manual ............... 1
Chapter 6. Cassette DEBUG User's Manual ............... 1
Chapter 7. Cassette EDIT User's Manual ................. 1
Chapter 8. Cassette ASM User's Manual ................. 1

ENCyclopedia Processor Technica
Volume 5
ALS-8

Contents

Chapter 1. ALS-8 Program Development System
Operator's Manual ........................................ 1
Change Notice #2: Revisions to Appendix C ........ 107
Chapter 2. ALS-8 Systems Group, Vol. 1, No. 1,
March 15, 1977 ............................................. 1
Chapter 3. ALS-8 Systems Group, Vol. 1, No. 2,
September, 1977 .......................................... 1

ENCyclopedia Processor Technica
Volume 6
PTC Memory Boards

Contents

Chapter 1. 2KRO ............................................. 1
Chapter 2. (This chapter reserved for future use.) .. 1
Chapter 3. (This chapter reserved for future use.) .. 1
Chapter 4. 8KRA ........................................... 1
Chapter 5. 16KRA & 32KRA ............................... 1
16KRA Manual ............................................. 73
32KRA Manual ............................................. 139
16KRA & 32KRA Updates ................................. 151
Trouble Shooting 16KRA & 32KRA ................. 153
Library diskettes are donated programs. They are generally in public domain, unless copyrighted by the author within the program itself. All diskettes 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.

---

**SOLOS**

A copy file for standard and my extended version of SOLOS. The original program is copyrighted free the CRW users of PDOS. Tech. Inc. has rights to this program. Functions for the input and output PROM board are not available for this diskette. (Donated by Ron Parsons.)

**SOLOS**

Further documentation of the files SOLOS(6), etc. above. (Donated by Ron Parsons.)

**COFFYS**

If you have them listed after command (separated by commas) from disk 0 to disk 1 preserving attachment. Requests permission to retrieve an existing file. (Donated by Ron Parsons.)

**RELCS**

Relocation program from July 1977 Byte.

**MFLES**

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

**PDISP**

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

**RENUM**

Removes file numbers (first five files) from named file. (Donated by Ron Parsons.)

**INTNLS**

Interrupt driven background electric driver. To be (?) described in PCTE ACCESS. (Donated by Ron Parsons.)

**SPFLES**

Display a compact list of all files on the unit given as parameters or a given unit, given default. Name stands for "Short Files". (Donated by Ron Parsons.)

**COMANS**

Command image doesn't work. File command above. (Donated by Ron Parsons.)

**PRGNS**

Standalone Cronesco Byteassembler pro-programmer (SOLOS) (Donated by Ron Parsons.)

**REOOG**

A PDOS disk reorganizer. Copies all files from unit zero to unit one. Does not rewrite existing files. (Donated by Ron Parsons.)

**CONWAS**

Compares the two PDOS files needed listing differences Assembly source file. (Donated by Ron Parsons.)

**SOL**

Assembly source code for PDOS command "SOLOS" which turns control over to SOLOS. Assumes SOLOS is at CCOO as in the PDOS. (Donated by Ron Parsons.)

**DSERV**

EDBASIC program for descriptive statistics (mean, etc.) (Donated by Stan Salkow.)

**PRINTER**

Driver for electric terminal (IBM 3270 compatible) on SOL's serial port. Uses SOL built-in DATA and PDOS driver area. BE SURE TO CHANGE USING 'SOL' UNDER OPTIONS. System reset will wipe out your installation part of the file. (Donated by Ron Parsons.)

**DOWES**

Device driver for delayed-mode subroutines. It moves bytes from a source address to a destination address. (Donated by Chuck Ellis.)

**LD**

List directory in alphabetical order, files name only. DOESN'T READ PARAMETERS; ALWAYS USE DEFAULT UNIT. (Donated by Chuck Ellis.)

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

---

**NEWFIT**

This is a corrected version of the GTF command that was originally released in PDOS 1.4. It automatically will find all files which are not present by the command. The files found by the command are those that are actually written on disk. You can copy NEWFIT to GET on your working diskette. (Donated by Processor Technology Corporation.)

**NEWFILES**

A corrected version of the FILES command which returns upper and lowercase names of files not present in files. The files released in PDOS 1.4 returned different names when searching for files that match bracketed substring specifications. This command and FILES SV, which are different names for the original command, but this version is corrected. (Donated by Processor Technology Corporation.)

**FOUR**

Generates random "four-letter words". Mode Select terminates and returns to PDOS. Words are displayed in large blocks of text. This version uses four-letter words. (Donated by Ron Parsons.)

**HELP**

Provides a brief explanation of PDOS commands. If a command file is not a part of the HELP command, a non-existent command name HELP, an explanation of the named command is generated. (Donated by Processor Technology Corporation.)

**HELP**

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

**MINDS**

Assembly language program for the MINOS Robot Control Language by Kenneth Long, see DR. Robotic Planner, Sept 77, revised by Ken Anderson, DH. ROBOS, May 78. (Donated by Ken Anderson.)

**PASCAL**

The Standard Pascal System, dated 5-13-78, from Stanford Linear Accelerator Center, Stanford University. (Donated by Richard Jahnke, Computer Science Group, S.L.A.C.) Essentially the entire P-code implementation for the IBM 360/70 computers, as implemented for the IBM 360/70 computers. Except for program declarations and passing PROCEDURES as parameters, it adheres to the standard Pascal defined by the standard commonly known as the N.U.C.C. system. The Pascal system under PDOS run, and thus it can pass data to other programs not written in PASCAL using normal PDOS files structure. This preliminary release did not have the REAL arithmetic implemented in the interpreter, so only INTEGER arithmetic can be used. REAL will compile. We expect to have the REAL arithmetic in a future library release. (Donated by Ron Parsons.)

**PASCAL**

See file PASDOC for more complete description. The source PASCAL compiler is on the PDOS diskette, and the assembly source for the interpreter is NOT needed. Sources are on PDOS for USER FILE DISK. (Donated by Ron Parsons.)

**PASDOC**

Documentation for the standard micro Pascal system. THE FOLLOWING FILES ARE RELATED TO PASCAL AND ARE DESCRIBED IN **PASDOC**: **PAS**, **PAS.PASC**, **PAS.PTS**, **PASCAL**, **PASDOC**, **PASCAL**. (Donated by Ron Parsons.)

**FOURDA**

An EDBASIC program for 2 video player football. Self-documenting. Source written by Larry Priske; adapted to disk BASIC by Stan Salkow.)

**NOTICE**

Important legal notices regarding this diskette.

**FEEDBACK**

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

---

**CONTENTS OF PROTEUS DISK 82**

**PASM**

Source code for the SLAC Pascal compiler, written in Assembly language. This should be loaded into the SLAC diskette after the background diskette on the diskette. Also see Processtex, or PLP, Vol. 2, No. 2 and following issues for more details on SLAC Pascal. (Donated by Stan Salkow.)

**PASM**

Source code for the prototype Pascal compiler, written in Assembly language. This should be loaded into the SLAC diskette after the background diskette on the diskette. Also see Processtex, or PLP, Vol. 2, No. 2 and following issues for more details on SLAC Pascal. (Donated by Stan Salkow.)

**PASDOC**

Documentation for the standard micro Pascal system. THE FOLLOWING FILES ARE RELATED TO PASCAL AND ARE DESCRIBED IN **PASDOC**: **PAS**, **PAS.PASC**, **PAS.PTS**, **PASCAL**, **PASDOC**, **PASCAL**. (Donated by Stan Salkow.)

**FOURDA**

An EDBASIC program for 2 video player football. Self-documenting. Source written by Larry Priske; adapted to disk BASIC by Stan Salkow.)

---

**NOTICE**

Important legal notices regarding this diskette.

**FEEDBACK**

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

---

**CONTENTS OF PROTEUS DISK 82**

**PASM**

Source code for the SLAC Pascal compiler, written in Assembly language. This should be loaded into the SLAC diskette after the background diskette on the diskette. Also see Processtex, or PLP, Vol. 2, No. 2 and following issues for more details on SLAC Pascal. (Donated by Stan Salkow.)

**PASM**

Source code for the prototype Pascal compiler, written in Assembly language. This should be loaded into the SLAC diskette after the background diskette on the diskette. Also see Processtex, or PLP, Vol. 2, No. 2 and following issues for more details on SLAC Pascal. (Donated by Stan Salkow.)

**PASDOC**

Documentation for the standard micro Pascal system. THE FOLLOWING FILES ARE RELATED TO PASCAL AND ARE DESCRIBED IN **PASDOC**: **PAS**, **PAS.PASC**, **PAS.PTS**, **PASCAL**, **PASDOC**, **PASCAL**. (Donated by Stan Salkow.)

**FOURDA**

An EDBASIC program for 2 video player football. Self-documenting. Source written by Larry Priske; adapted to disk BASIC by Stan Salkow.)

---

**NOTICE**

Important legal notices regarding this diskette.

**FEEDBACK**

Explanation of how to report problems you encounter with this diskette's programs.
It assumes a Courier 10 prntwhei, or equivalent arrangement. It is based on the IBI8printer2 driver, with references to the extra features of the IBI8printer interface deleted. (E Sokolow)

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

MONTST Documentation of MONTST.

OKDATA Source code for PDOS driver to operate OKdata printer on serial port of Cramenoc 600-ART interface board.

PRINT Centronics 701 driver. [Donated by Roy Heybuck.] PRINT Driver for an old communications printer. [7] [Donated by Roy Heybuck.]

PDOS A printer driver for a Selectric terminal IBM 2741-type. Correspondence coded, uses the Sol serial port, modified to do 124.2 baud. [Stan Sokolow]

RETIR An updated version of RETIR on 8k disk to change graph headers. Do not set up right. [Donated by Roy Heybuck, CPP.]

Mastes3 This is a dual masterboard game. Two people play at the same time on different game boards. The game has a mere of numbers within the matrix, automatic return, and a more than one game set to determine the winner. Each player enters his turn and tries to take as much area as possible, i.e. the first player enters a queen and a cannon. The second player writes his queen on the other side of the board and the machine automatically continues without the need for the carriage return. (Michael Richardson)

CALC2 I teach Mechanical Technology at Saunders Technical High School in Yorkville, This program will print and read from the matrix. That's nice. Abandoned the program is that the calendar is general ized so that any year of the calendar is used, and an accurate calendar will be printed. There is also an interesting generalized, (This is what I was doing in the middle of the picture and the calendar printing) (Michael Richardson)

SOL, 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)

MULTI3A A multiplication practice program. (Michael Richardson)

MULTI3B An addition practice program. (Michael Richardson)

W1 A program that I use whenever I go on a diet. The data is recorded in a file after the program. Each time a new diet is entered the program expects to get the day number, the food, the calories, and the weight at the beginning of the program. The weight is updated for all other entries on that day. There are alphabetical calorie counts for some of the foods. The program writes into the program in the format.

STT The program I use to automatically right justify a letter. (Michael Richardson)

T7 The program I use to print a file I've already right justified. It is in protected (I attribute). (Mike Richardson)

CMPP Prepared for comparison command, in assembly source code. See the program comments. (Tom M. Quinn)

MENKB,A Memory back-select, Crornenoc-style. See the source code. [Tom M. Quinn]

BAUPD A command to select baud rate on PS3 board. [Tom M. Quinn]

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

SETUPD Command to initialize TI B10 printer. [Tom Quinn]

TIIID TI B10 printer driver based upon Sol Printer 3 driver. [Tom M. Quinn]

SORTF Shell sorting in PDOS-FORTRAN. [Earl Dunham]

SORTB Same as SORTF, but written in BASIC. [Earl Dunham]

SPECIAL 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 PROCESS. This file contains commands that are used in PROCESS and also doubles as an example of the use 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.

LDevice driver for parallel printer. (Michael Richardson)

PROG3A Explains the programs submitted by Michael Richardson.

STARTER3 Explains the starter programs by Michael Richardson.

EDIT3A A brief explanation by Michael Richardson of why he renamed the PDOS editor from EDIT to E. It also has a brief synopsis of the EDIT features.

BOOT3S The source for the Helios bootstrap loader.

COLP3A A screen doodling program which places a duplicate copy in memory for saving. Good for designing playing fields for video games.

DESFORTH A For the Sol ports serving Helios. Gives port definitions and bit addresses for all the ports used by the Sol for communicating with Helios.

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


MSTST3S A collection of memory test programs tailored for the various boards produced by PCT. [J. Maguire] [Editor's note: This MSTST is not same as MSTST on this disk]

OCTAL3S An Octal Enter and Dump routine. Gives ASCII values and a Hex address for reference.

PARASIS In the source code for Lichen Wang's Palo Alto Tiny Basic from early issues of Doctor Dobbs' Journal. A few enhancements may be necessary to reconvert to a random file if desired.

SPINN3S NEC Splawriter 5510/5520 device driver. A di-directional, logic seeking printer driver fully compatible with wordprocessing. Also includes "space averaging" as ability to restructure the line producing "typewriter" quality printing. [Tom M. Quinn]

TERM3S The missing TERM command from Sol Bootload proc.

ZAPI3S Zip it let's you defeat the attribute protection of PDOS. UNFAIR yets you back. Use with caution!

The New and Better B-4 Patch 1 is a program to allow the recording of attribute protected files on PDOS 1.3 (not 1.3). [See test file HOWDY for more information on operation.]

MESSAGE, MESSAGE, BASE, RES, RES, TEMPI a group of programs that were done for Jon. Studying the source code is a good way to learn and understand how to use the overlay handler and how to interface to PDOS in general. See test file HOWDY for more information on operation.

PRIM3E is a friendly prime number generating program. See test file HOWDY for more information.

FACTOR is a program to factor into an integer into prime. See test file HOWDY for more information.

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

QUIEST same as above but with modifications suggested by W.B.

SHEARS From Knuth

SORT3S From Knuth

SORT is another sort suggested by Knuth which seems to approach the speed of the gnu sort but may not have the same access for use. For example Knuth rates the mathematical evaluation of this sort 3.7 points, his maximum.

MAE 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 word encoding, pharmaceutic, etc. to see it work. (Should be entered in lower case)

FIND3S was originally inspired by the IEEO micro-srobe contest. This was the best program I came up with, and it's basically using a recursive function. Note that it will require lots of memory to run SET your buffer > 3000 and use the MRASS on this disk.

FIND3S is another mouse program but with a unique idea that causes interesting behavior sometimes. Consider an area where the walls are at 999 and the corridors are set at 0 and the "corridor" is set at -1. Here 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 FIND3S. Watch its behavior in lower case.

PERM3S 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 BANNERS.

BANKERS is a program to enter deposits and withdrawals, and obtain balance of total and percentage.

CREDIT will create random file of accounts. Used with BANNERS program.

EXPENSES is a sample list of expenses for use with BANNERS.

PERSONAL is a program to create serial files of names or titles that may be converted to a random file if desired using SERKRM program. For use with BANNERS.

SERKRM is a program to convert serial files to random access files for use with BANNERS.
BIOPLOT

A competing version of the biophotry plotter, this can use the Woodard print drivers. If the basic
is included with the matrix operations deleted, this program can be used to print on any of the
printers listed above, as well as to text files on the disk.

BIOREAD

This is a text file used by BIOPLOT/5/BIOREAD. This file contains the names and descriptions of all the
commands in the biophotry plotter. Other pertinent information is also contained in this file.

BIOREAD

A basic program that converts binary and leadership

Simulating equations.

SIMULS

SIMUDS

SIMUD

The data file for holding equations to be solved.

H - 7

This file lists the contents of this diskette, H-7 from the Prestro 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 see the file WARRANTY and

Feedback.

Charles L. Ahney, III
Prestro Library

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

CALCULATE YOUR ACCOUNT BALANCES AND PERCENTAGES

CREATEP - CREATE RANDOM FILE OF ACCOUNTS

EXPENSES - SAMPLE LIST OF EXPENDITURES

PERSONAL - PROGRAM TO ENTER DAILY OR WEEKLY EXPENSES, AND

OBTAIN REHEAT OF TOTALS AND PERCENTAGES

SERIAL - PROGRAM TO CREATE SERIAL FILE OF USING ER.INDR PROGRAM

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

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

PATCHLS is a program to allow the reattributing of attribute

protected files on DOS.PDS/5 (but not D/4). I don't have

the source code unfortunately but it is pretty simple.

Type PATCHLS and it will execute and return to

PDOS. Then REATTR (the files). For safety, you should reattribut

PDOS will continue to ignore the attribute protects.

MESSAGE, MESSAGE, BUILD, MES.S, and MES.TEMP are all a group of

useless programs I did for fun. Studying the source is

good way (maybe) to understand how to use the

overlay handler and how to interface drivers in general.

MESSAGE will type a random (almost) message on the screen

whenever run. I use it in my START.UPI file to avoid

the same old boot up and over and over again.

MESSAGE is a utility file that contains the 32 messages

that MESSAGE may choose from. MESSAGE.S 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 code for a message.

FORMAT is the text formatter originally described in "Estimated Tools" by Kernighan and Plauger. It was
written by Mike Gabrielson and printed in the May 79

"This is the contents of this diskette, H-7 from the Prestro 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 see the file WARRANTY and

Feedback.

Charles L. Ahney, III
Prestro Library

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

CALCULATE YOUR ACCOUNT BALANCES AND PERCENTAGES

CREATEP - CREATE RANDOM FILE OF ACCOUNTS

EXPENSES - SAMPLE LIST OF EXPENDITURES

PERSONAL - PROGRAM TO ENTER DAILY OR WEEKLY EXPENSES, AND

OBTAIN REHEAT OF TOTALS AND PERCENTAGES

SERIAL - PROGRAM TO CREATE SERIAL FILE OF USING ER.INDR PROGRAM

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

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

PATCHLS is a program to allow the reattributing of attribute

protected files on DOS.PDS/5 (but not D/4). I don't have

the source code unfortunately but it is pretty simple.

Type PATCHLS and it will execute and return to

PDOS. Then REATTR (the files). For safety, you should reattribut

PDOS will continue to ignore the attribute protects.

MESSAGE, MESSAGE, BUILD, MES.S, and MES.TEMP are all a group of

useless programs I did for fun. Studying the source is

good way (maybe) to understand how to use the

overlay handler and how to interface drivers in general.

MESSAGE will type a random (almost) message on the screen

whenever run. I use it in my START.UPI file to avoid

the same old boot up and over and over again.

MESSAGE is a utility file that contains the 32 messages

that MESSAGE may choose from. MESSAGE.S 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 code for a message.

FORMAT is the text formatter originally described in "Estimated Tools" by Kernighan and Plauger. It was
written by Mike Gabrielson and printed in the May 79

"This is the contents of this diskette, H-7 from the Prestro 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 see the file WARRANTY and

Feedback.

Charles L. Ahney, III
Prestro Library

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

CALCULATE YOUR ACCOUNT BALANCES AND PERCENTAGES

CREATEP - CREATE RANDOM FILE OF ACCOUNTS

EXPENSES - SAMPLE LIST OF EXPENDITURES

PERSONAL - PROGRAM TO ENTER DAILY OR WEEKLY EXPENSES, AND

OBTAIN REHEAT OF TOTALS AND PERCENTAGES

SERIAL - PROGRAM TO CREATE SERIAL FILE OF USING ER.INDR PROGRAM

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

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

PATCHLS is a program to allow the reattributing of attribute

protected files on DOS.PDS/5 (but not D/4). I don't have

the source code unfortunately but it is pretty simple.

Type PATCHLS and it will execute and return to

PDOS. Then REATTR (the files). For safety, you should reattribut

PDOS will continue to ignore the attribute protects.

MESSAGE, MESSAGE, BUILD, MES.S, and MES.TEMP are all a group of

useless programs I did for fun. Studying the source is

good way (maybe) to understand how to use the

overlay handler and how to interface drivers in general.

MESSAGE will type a random (almost) message on the screen

whenever run. I use it in my START.UPI file to avoid

the same old boot up and over and over again.

MESSAGE is a utility file that contains the 32 messages

that MESSAGE may choose from. MESSAGE.S 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 code for a message.

FORMAT is the text formatter originally described in "Estimated Tools" by Kernighan and Plauger. It was
written by Mike Gabrielson and printed in the May 79
The following programs were donated by Jay Parson of Somerset Data Systems, Inc. Bernardsville, NJ.

ROBOTS is an old game involving hiding from killer robots.

The following program to help convert between CFM and PTDOS format files was donated by L.B. Bardenbry, Greenlake, W. I.

The following program was donated by Earl J. Hearn of La Noba, CA.

HELLOS LIBRARY DISKS 4

HELLOS Library Disk No. 8
April 16, 1981

This disk contains the small C compiler as implemented by Tom Cain and enhanced by Ed Lirgelt.

C - The running compiler.
C.C - C source for the compiler.
C.C - DOS ASM source for the compiler, the results of compiling the compiler.
C.R - The runtime support package for the Sol-Hellos system.
C.C - The common definitions needed by the compiler.
C.X - A description of each routine in the compiler.
C.S - A sample C program.
C.S - A sample C program.
C.S - A sample C program.
C.S - A sample C program.

C.R - The runtime support package for the Sol-Hellos system.
C.C - The common definitions needed by the compiler.
C.X - A description of each routine in the compiler.
C.S - A short description of how to use the compiler.
C.S - A sample C program.

Please address any questions to Chuck Abbey (415) 419-3377, 537 Shokanabum Circle, Livermore, CA 94550.
A SIDE BENEFIT TO ELECTRIC PENCIL VERSION SS
by H. Leon Winter

I have some more notes 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 benefit came by surprise. That was reverse video in the 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 typist. 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 be ored with 80H stored, this will continue until the MODE SELECT key is pressed a third time which causes 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 function at all times. A small change in the program could modify this.

I had to get by brute force on this one. The first was to find a way to get a control character in the first place. EP uses every control character available to perform immediate control action. So I needed some way to stop this. This 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 UDM. Many of these have an effect on the display on the screen. I wanted 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. Everything something is moved in the buffer, which is practically constantly while inputting text, all the text gets shoved through a ANSI 7TH shaped hole which unmask 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 occurs.

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 some 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 in addition a 36h for future use!

The following locations in version SS need to be changed to NOP's. Each address pair had been an ANSI 7TH before.

Location 0090H had been CD 39 09 which called one of EP's keyboard input routines. Change this to CD 23 0D which calls the start of the new control character routine.

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

<table>
<thead>
<tr>
<th>Location</th>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0021h</td>
<td>54 20 52 45 41 44 00 52 00</td>
<td>T READ R</td>
</tr>
<tr>
<td>0028h</td>
<td>54 20 53 41 56 45 00 57 00</td>
<td>T SAVE W</td>
</tr>
<tr>
<td>002Ch</td>
<td>57 4F 52 44 20 23 00 58 00</td>
<td>WORD # X</td>
</tr>
<tr>
<td>0032h</td>
<td>52 43 52 44 20 23 00 59 00</td>
<td>RERE # Y</td>
</tr>
<tr>
<td>00A8h</td>
<td>43 20 40 41 46 54 00 43 41 41 00</td>
<td>C AFT CAA</td>
</tr>
<tr>
<td>00B1h</td>
<td>43 20 42 46 52 40 00 43 4F 42 00</td>
<td>C BFR CAB</td>
</tr>
<tr>
<td>00CC4h</td>
<td>43 4C 20 53 59 53 00 43 4C 52 00</td>
<td>CL SET CLR</td>
</tr>
<tr>
<td>00CC7h</td>
<td>4A 53 59 54 46 59 00 4A 30 20 31 01</td>
<td>JSTIFY J0-1</td>
</tr>
<tr>
<td>00CD4h</td>
<td>4C 45 20 53 50 48 00 53 81 20 35 01</td>
<td>LF SPC 5L5</td>
</tr>
<tr>
<td>00CF1h</td>
<td>50 47 20 4C 45 44 00 47 2D 37 32 00</td>
<td>PG LEN G-72</td>
</tr>
<tr>
<td>00CF6h</td>
<td>4C 46 54 20 4D 52 00 4D 30 20 00</td>
<td>LFT MR MO-</td>
</tr>
<tr>
<td>00C02h</td>
<td>50 41 4F 45 20 23 00 4E 31 23 00</td>
<td>PAGE # 23</td>
</tr>
<tr>
<td>00C0Ch</td>
<td>50 52 52 20 4C 44 00 50 30 20 00</td>
<td>FKT LN PO-</td>
</tr>
<tr>
<td>00D17h</td>
<td>4C 4E 45 20 4C 44 00 4C 32 35 20 00</td>
<td>LINE LN L-25-</td>
</tr>
<tr>
<td>00D23h</td>
<td>SPACE AVAILABLE FOR ROUTINES = 90 BYTES</td>
<td></td>
</tr>
<tr>
<td>00D78h</td>
<td>START OF NEXT MESSAGE FILE</td>
<td>TIME</td>
</tr>
</tbody>
</table>

The sources 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 be handy for a Hyper type driver I've got on the back burner. My regards to you, Stan, and to the members.

Sincerely,

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

ASSM 0D23 5000

0D23 1100 CONTROL CHARACTER INPUT ROUTINE FOR ELECTRIC PENCIL
0D23 0D23 1101
0D23 0D23 1110 ALLOWS CONTROL CHARACTERS TO BE "HIDDEN" IN TEXT
0D23 0D23 1120 ALLOWS REVERSE VIDEO CHARACTERS TO BE IN TEXT ALSO
0D23 0D23 1123
0D23 0D23 1140 WRITTEN BY H. LEON WINTER FEB 1982
0D23 0D23 1150 AT NASULI, MALAYALAY, BUKIDNON, PHILIPPINES, 8201
0D23 0D23 1160
0D23 0D23 1170 EQUATE TABLE
0D23 0D23 1180 KYBD2 EQU 059H ONE OF EP'S KEYBOARD ROUTINES
0D23 0D23 1190
0D23 0D23 1200 IN ADDITION TO THE CHANGES NOTED IN THE DOCUMENTATION,
0D23 0D23 1210 WHICH INCLUDE THE NEW COMMAND SUB SET MSG DATA AND THE
0D23 0D23 1220 ADDITION OF THE MODE SELECT KEY THAT IS REPLACED WITH NOPE. THERE IS THE
0D23 0D23 1230 INITIAL CALL TO THIS PROGRAM, THE LINE OF CODE AT 0090H
0D23 0D23 1240 IN VER. 9 OF EP MUST BE CHANGED FROM CD 39 09
0D23 0D23 1250 TO CD 23 0D
0D23 0D23 1260
0D23 0D23 1270 CTLIN ORG 0D23H 1ST FREE BYTE AFTER SUB SET MSG
0D23 0D23 1280 CALL KYBD2 CHARACTER USUALLY INPUT FROM HERE
0D23 0D23 1290 MOV A,C EXAMINE CHARACTER
0D23 0D23 12A0 CPI 7FH IS IT A signalling KEY?
0D23 0D23 12B0 RE , MAINTAIN DELETES FUNCTION
0D23 0D23 12C0 RE , MAY BE A LINE FEED
0D23 0D23 12D0 RE , IF SO, MAINTAIN IT ALSO
0D23 0D23 12E0 LDA FLAG SEE IF IN SPECIAL MODE OR NOT
0D23 0D23 12F0 ORA A IF FLAG IS 0 ALL IS NORMAL
0D23 0D23 1300 ORN OLESET IS SET FOR SPECIAL MODE
0D23 0D23 1310
0D23 0D23 1320 NOT IN SPECIAL MODE IF HERE
0D23 0D23 1330
0D23 0D23 1340 MOV A,C EXAMINE CHAR THAT WAS INPUT
0D23 0D23 1350 ORA A CHECK FOR MODE SELECT KEY
0D23 0D23 1360 ORN OLESET COMES THROUGH ANSI 7TH IN KYBD2
0D23 0D23 1370 SET STILL NORMAL MODE, PUT CHAR TO CRT
0D23 0D23 1380
0D23 0D23 1390 IF HERE THEN MODE SELECT KEY WAS INPUT FOR MODE CHANGE
0D23 0D23 13A0
0D23 0D23 13B0 MODE1 MVY A,01H SET UP FOR MODE CHANGE
0D23 0D23 13C0 STA FLAG FLAG NOW REFLECTS CHANGE
0D23 0D23 13D0 CALL KYBD2 GET CHARS TO BE CHANGED
0D23 0D23 13E0 MOV A,C CAME BACK IN C
0D23 0D23 13F0 ORA A 0 IF ANOTHER MODE SELECT

CON'T ON PAGE 16
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 Shreder didn't sell rights to the Pencil instead of allowing it 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 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 teletypewriter, not for those which have the custom printer drivers. Hence, before changing any code, one should check version numbers. These are found at addresses 110-11F hex. Also, since there may be several editions of this version one should also check to determine if the id 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 60 codes.

Sincerely yours,

March 7, 1982

Louis T. Bucklin

(Ed. note: I had an interesting conversation with someone who says he testified as an expert witness in a lawsuit between Michael Shreder and the Vector Graphic computer company. It seems Shreder 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 Shreder's code were identical to Processor Tech's code in the Software #1 editor, so Shreder's case was dropped down. (I don't have the right author) and to avoid counterclaims. That's why Electric Pencil disappeared from the marketplace, the story goes.)

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

<table>
<thead>
<tr>
<th>Addr</th>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>171</td>
<td>3E</td>
<td>D6C</td>
</tr>
<tr>
<td>170</td>
<td>0D</td>
<td>00</td>
</tr>
<tr>
<td>171</td>
<td>0D</td>
<td>E3A</td>
</tr>
<tr>
<td>179</td>
<td>FD</td>
<td>ED</td>
</tr>
<tr>
<td>26A</td>
<td>FX</td>
<td>FD</td>
</tr>
<tr>
<td>28A</td>
<td>CC</td>
<td>F1F</td>
</tr>
<tr>
<td>3C0</td>
<td>FF</td>
<td>F39</td>
</tr>
<tr>
<td>3D6</td>
<td>CF</td>
<td>FF</td>
</tr>
<tr>
<td>515</td>
<td>22</td>
<td>FD</td>
</tr>
<tr>
<td>516</td>
<td>2C</td>
<td>FB1</td>
</tr>
<tr>
<td>646</td>
<td>FF</td>
<td>FFB</td>
</tr>
<tr>
<td>656</td>
<td>0D</td>
<td>FBF</td>
</tr>
<tr>
<td>660</td>
<td>102A</td>
<td></td>
</tr>
<tr>
<td>7C0</td>
<td>FD</td>
<td>F11D</td>
</tr>
<tr>
<td>984</td>
<td>0D</td>
<td>1108</td>
</tr>
<tr>
<td>A10</td>
<td>FD</td>
<td>1148</td>
</tr>
<tr>
<td>A20</td>
<td>FD</td>
<td>0D</td>
</tr>
<tr>
<td>A58</td>
<td>0D</td>
<td>1108</td>
</tr>
<tr>
<td>A5E</td>
<td>FD</td>
<td>120D</td>
</tr>
<tr>
<td>A62</td>
<td>FD</td>
<td>123B</td>
</tr>
<tr>
<td>A65</td>
<td>CE</td>
<td>F28</td>
</tr>
<tr>
<td>ACO</td>
<td>FD</td>
<td>123D</td>
</tr>
<tr>
<td>3B2</td>
<td>FC</td>
<td>123F</td>
</tr>
<tr>
<td>D43</td>
<td>FC</td>
<td>1304</td>
</tr>
<tr>
<td>D69</td>
<td>FC</td>
<td>130F</td>
</tr>
</tbody>
</table>

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 runs all software at F000 hex, the time and potential error of switching becomes too burdensome.

I made considerable effort to contact Michael Shreder 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 would benefit from error removal and extension.
HELP NEEDED ON 1200 BAUD TAPES, G2 MICROSOFT AND MCKELVEY' MCP
by Alastair Preston

I have several questions that readers of PROTON/NEWS may be able to answer
for me. As background info, I am running an S100 system under CUTER (ROM) with
a 256 Kpu, 32K memory, IBM65, CUPS and a TV18 video board set at 400M.

1) The CUTER 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 P000, nothing has been said
about CUTER -- are replacement ROMs available to put CUTER at P000 ?

3) I have both 8D_BASIC and G2 Microsoft Basic; has anyone implemented the
8D_BASIC cassette routines in G2 ? (i.e. FILEM, MKWIN, READ, MKWIN and
CLOSEB). MICROSOFT does not have the source code for G2 BASIC and are
unable to help.

4) In implementing R. McKelvey's "Micro Communication Package" (F/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 ASCII and EDIT by reassembling the program
with an extra INC H to skip one character at the beginning of each line,
and editing the line numbers in the receiving system's file. Any attempt
to send a BASIC listing bombs as the number of characters preceding the line
number is not constant. Has anyone solved this? Also, since I do not have a
diskette controller presents some problems. I would like to use cassette
instead. My attempts to make use
of the rather limited information in the CUTER and CUPS manuals have been
unsuccessful; I just haven't 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 P000 be available
on standard 5 1/4" disks? [Ed. note: Not in our plans.]

6) Does anyone know of an available Z80 assembler/editor (256k monomos) on
CUTER tape; also a usable database and any Adventure games?

Sincerely,

15 D Twin Terrace
Edmonton, Alberta
Canada, T6E 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 themselves. I imagine
due to the length it was impossible. I received several requests for North*Star Single Density disk copies as well as
Double Density. I am happy to provide one 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,

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 B28 which is dedicated to
handling the video function, it offers all 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
256A processor, 4MHz
512 user programmable characters (128 defined on
initialization, can be redefined)
4K EPROM
16K RAM
HI-RES graphics - 640X288 (implemented with the
programmable character generator)
Keyboard port + type-ahead buffer
Centronics printer port
Light pen interface
RF shielding on all high frequency components
Supercrits, subcrits, underline, strike-thru,
blanking, half-intensity, flashing, inverse, etc.

The initial configuration is for an S100 video card.
However, since I have two SOLs 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 present 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 4MHz 7800 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 expect the latter to
sell at a price comparable with the Micro Complex board.

Regards

Michael Carter
4 Richardson St
Darren ACT 2605
AUSTRALIA

PS. For three years, I have puzzled over why SOLOS was
located at C000 rather than (eg.) at P000. My partner has
suggested a reason: C000 makes it simple to implement the
PHANTOM with an Exclusive OR.

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

After adding a Micropolis Mod II disk drive to my SOD 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. MBASIC has some nice features: MBASIC supports random access disk files, can achieve up to 60 digital counts of accuracy if desired, and has nice program tracing. On the other hand, ECBASIC supports IF..THEN..ELSE, has a goto 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/MBASIC 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 4000H for any hex digit XX and move the code to its new location(111,276),(717,646). I call this relocated disk operating system MDOS. It occupies 2B20H to 8800H.

Needed next was a disk executive for ECBASIC. SDEXEC sits above MDOS and requires 512 bytes of memory. Upon initialization SDEXEC modifies four 1/0 vectors in ECBASIC and some code in MDOS. At this point all input goes through ECBASIC which is completely transparent to ECBASIC. The executive takes control however when it sees CONTROL D and prompts for a disk command.

The commands are:

DOS (to exit ECBASIC and enter MDOS)
SOD "parameters" (to save a program on disk)
FEED "parameters" (to load a program from disk)
REPLACE "parameters" (to update a file disk)
RENAME (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 side from the keyboard or as the last instruction of the program. By using source code the programs are more portable. A BASIC program written in the correct syntax of both BASICS 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. Microcode LISTed while connected to a card-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 tape, please send $15 for MDOSMOVER, $10 for SDEXEC or $20 for both.

Ron Shenk
329 Robin Hood Rd.
Atlanta, Ga. 30309
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 DJ2DB11 locates its bootstrap prom and ram from $0000H$ 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 MSCOLOS 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 DJ2D11. 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 Protex News, Vol. 4, No. 5/6, p.1) about being able to "turn off" MSCOLOS and MCDVM by outputting 1s to the two high order bits of port 8F. That gave me my "brilliant" idea. Perhaps I could move the DJ2D11 to F000H thus leaving room for a 62k CP/M system, and only turn on MSCOLOS and MCDVM when I wanted access from the keyboard 0 to the screen. This would leave me with a somewhat complex bootstrap procedure but would gain me 6k of space for my system.

I quickly called Morrow about a "Relocation Package" for the controller 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 to you 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 MSCOLOS 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 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 MCOLOGS and MCVD on an internal bus, separate from the 100 bus, they did not mean that the signals from the two busses did not interfere with each other at the same address location. I assumed that MCOLOGS was interfering with the controller located at the same address. Sure enough, when I entered and executed a short program to turn off MCOLOGS, 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 MCOLOGS was on and on when MCOLOGS 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 JA and JB 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 unchanged. This change turns the board off at power up and on 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 later. (3) Turn paddle 7 of switch 1 the "on" position (Switch 1 is located at 50). This change inserts one wait state in bus cycles which read from the 2708 ROM on the DJ2DB controller board. The original DJ2DB produced no error. This change says this must be off for 2MHz operation and must be on for 2MHz operation. Why is it 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 MCOLOGS 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 F000 to CXXX to run with MCOLOGS at all. In addition I had to insert code like the following to turn off the DJ2DB and turn on MCOLOGS 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 STDP and output to SOUT. Changing those special routines and making one additional change in the whole boot procedure is all that is necessary, 43 bytes total. In CTTY, 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 0ffh ;turn on mcosols by a/o p 0 in bits 6 & 7
```

```
call sout ;of 0ffh
```

```
mvi a, 0ch ;turn off mcosols by a/o p 1 in
out 0ffh ;bits 6 & 7 and on dj2db by
out 040h ;p/ 1 in bit 7
ret
```

In CTTY similar code should surround the call to sain as follows:

```
CTTY xra a ;turn dj2db off and mcosols on
out 040h ;
out 0ffh ;
call sainp ;get character from keyboard
push psw ;
save character
mvi a, 0ch ;
turn mcosols off and dj2db on
out 0ffh
out 040h
pop psw ;

turn character
```

(continue balance of CTTY as written)

Change CTTY beginning at CONIN3 as follows:

```
CONIN3 out 040h ;turn dj2db off and mcosols on
out 0ffh
```

```
conin3 call sainp ;get a character from keyboard
```

```
ja conin31 ;wait until there is one
```

```
push psw ;
```

```
save character
```

```
mvi 00h
```

```
out 0ffh
```

```
out 040h
```

```
pop psw ;
```

```
; retrieve character
```

(continue with existing code)

The only other issue is how to get from MCOLOGS at F800H to DJ2DB at F800H to cold boot the CP/M system. I fixed this with a somewhat inconvenient kludge. There is plenty of room in the MCOLOGS 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 MCOLOGS, type, EH 00cr. 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 00cr or WBXcr. If everything is right the 9 byte program will turn off MCOLOGS, turn on DJ2DB, and jump to DJ2DB's cold boot routine. Thereafter, the switching of MCOLOGS 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 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.

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 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 TMWNET, two telephony carrier companies. The auto-log feature checks the carrier detect signal from a modem to determine when to start the log-on sequence. TELENET 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 original version of this program was written in 2-80 on my S-100 computer. Since I don't have a SOL, there is 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 2-80 monitor, I believe that I understand the necessary protocol for I/O operations via SOL (I also had a source listing of SOL 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 TMWNET as your telephone network, you will have to set the conditional assembly flags that are near the end of the TELECOM:S file. Set the appropriate EQUATE to 1 and the other EQUATE to 0.

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 TELCM2: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 TELECOM: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 1/0 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 keyboard 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 on 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 once while you are transmitting, then press CTRL-Q to cause it to resume.

The menu option are:

1. Connect to The Source
2. Manual log-on
3. Change Source password
4. Quit and 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 you are running. THE NAME OF THE OBJECT FILE FOR THE PROGRAM YOU ARE RUNNING MUST HAVE BEEN ENTERED AT 'SIGNAM' 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 4CO. 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 transferred to PTDDOS. Since PTDDOS also maintains a buffer, the transfer may not always result in an actual disk write taking place. After the buffer has been reassembled, 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 begin (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 any control characters, object files cannot be handled. The annual 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 (PTDDOS apparently ignores it as long as you are not in PTDDOS). It is also used to abort the auto-log feature in case of problems; it skips to the main terminal where 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.

ESC: 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
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 puts the address of the 8080 and puts the instruction
that corresponds to the request into the 8080 via the data bus.
The BRST command in 8 types (RST 0 through RST 7) which acts
like one-byte call instructions to predetermined addresses to perform
task. The priority is determined by which pin it is assigned to, which
can be handled by the 8214. The highest priority first (determined by
which pin it is assigned to), which can be handled by the 8214.

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

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 of what the controller
wants).

We also also 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
computer 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, disabling 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 set the 8080's program set threshold).
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 tricky to install.
In any case, the SOL has to be modified by installation
of jumpers as described in the PDC update for vectored interrupt.
The SOL will prevent the onboard address decode 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 characters 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 re-read it. This is always a good procedure when any trouble is encountered. The time I will spend re-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. Ultimately 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 the computer would reset itself. (I had changed my keyboard some 2 years before so the REPEAT and BREAK were the resetting keys. It being me to have to re-arranging the Alpha Shift key after every reset! The BREAK key is also on pin 4 of U-22 hence 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. When it turned out, this feature was a help that I would not have known before this particular fuss. Going along the input pins of U-22 with the prob in the TTL position, I could find nothing. Now all the lines read high as they should if all is well. But all was not well so I thought 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 bright 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 3k ohm resistors. 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 places where there was a patch of corrosion 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, just as). When I cleaned those, 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 situation and do a good cleaning of the keypads on the circuit board. I use clean gauge and rubbing alcohol. Then dry the surface off with another clean gauge. Carefully reassemble the board, when all the screws are started, tighten them gently 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

Summer Institute of Linguistics
Neum, Malagapay
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 page 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 paper about the IBM-XX multiplexer, and one was an article about the IBM-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 two dynamic memory boards. In response to a message from Bruce Diller and his keyboard fix, which is a suggestion which 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 reading X-25 of the SOL Manual, it indicates that the signals are capacitively coupled, (even your fingertips will make the circuit work) so by shorting the circuitry repeatedly, one may be encouraging component failure. I have been filtering 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. 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, the keyboard customer 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 MC6800, NIDisk, The Last Memory

Dear Stan:

I just received Vol. 4 #5/6 of PROTEUS NEWS with the renewal note for 1982. 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 Microplex's Dual Personality Module. Not in its "standard" configuration with the 16K RAM/ROM option, but with SOL's dedicated address space, plus additional 8K bytes of PROMs. This required a rewrite of Microplex's SOLMONITOR and a revision of the wiring changes on the PCB. My reason for doing this was to make the SOL-20 usable as a single-board computer. The Dual Personality Module increases the SOL-20's memory to 32K bytes by using the SOL's unused address space. Actually, there is no space. Actually, there is no space. Actually, there is no space.

I also have the revised PROMS for the single-board computer. I was successful in getting PROMS from North Star's choice of ways to save. I removed the E000 - E0FF for the single density disk controller. While the price was still the reasonable $25, I purchased a set of PROMS for the range F000 - FFFF and used the range E000 - EFFF for the two 16K RAMs. This is the next best solution 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 booting and reassembling the entire system in assembly language compatible with the INTEL 8080A. The pay-off was relocatability and the fact that new DOS commands can be added without recompiling the system. This version of DOS performs the disk management functions, namely, zeroing a specified disk drive, reading a specific sector on a specified drive and writing a specific sector on 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.

I have removed a lot of old SOLs into RAM at C000 from old PROMs. This results in a lot of unused memory space. In fact, I have removed the 2716 containing the old version from Micro's ROM module to reduce unnecessary current drain from the battery. I have also removed the unused tape routines from old SOLs and tuck them into some space I have available on EPROMS above D000.

This 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 work with microfilms, I added a 470V/9 via a 340 74LS00 ULN gate to the 74LS300 ULN gate to the Microline printer. Some fix was necessary in order to avoid taking the metal caps off the SOLS 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 speeds (1000 baud) of the Microline printer was not readily available. The solution was to bring three leads out to a single-pole-double-throw center-off switch on the wall via three unused pins on jack J1 of the serial port. This joined my panel of switches to either inhibit the transmission of data to the 74LS300 ULN gate or the Microline printer. At the same time, software changes to NS BASIC and my assembler in order to take advantage of the new configuration were required. The switch was interfaced to the RS-232 interface as suggested. Now the jump to the top of the next page after printing 62 lines is automatic.

My 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 2616 disk drive in place of the 16K static RAM's which are replaced by 2716 EPROM's. The organization of the board will be as follows:

- E000 - CFFF RAM with 26 on-board 2616 memory chips
- D000 - DFFF ROM with 2 on-board 2716 EPROM's
- E000 - EFFF 2 empty sockets. Space for SOLS PROM
- F000 - FFFF 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 memory space. By design of the board, an empty socket will have no address space. Actually, there is no space. Actually, there is no space.

I can find good use for the 3K from F000 to FFFF for debugging routines. I have proposed to Static Memory Systems that the board be marked the 1K half (F000-FFFF) or the 2K half (FC00-FFFF) of the 2716 that conflict with the controller with FFS's. They agreed their "FFS detector (a 74LS30 IC) would not discriminate between these files. If one is occasioned to come from computers, they 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. 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,

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 board from Static Memory Systems arrived Feb. 9, 1982. It has been installed in my SOL and works fully. The EPROM board now contains only the North Star controller and the last memory board. The higher address half of the 2716 EPROM board overlaps the address space into which the NS 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 FF bytes.

With this board, 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 comment would not be shared by many. I would have preferred that Static Memory Systems used "port addressing" for bank selection as does Goddard on the XIII memory board rather than their method of decoding the eight 8-bit 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 on the portion (16K) of my Goddard board which switches the address data bus to the address data bus to the Microline printer. Using the port addressing, I could plug the printer board into the address data bus on the top of the backplane and my printout would send the disabling byte to the printer via the 88XX OUT instruction prior to beginning the print sequence. I can now 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/enableing a segment of the last memory board via one of the unused extended memory lines on the S-100 bus. I have to know how to do it, but that's the fun of having a personal computer not made unchangeable by IBM.

CONT 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 1500 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,

Robert C. F. Bartels

P.O. Box 2240
Ann Arbor, MI 48106

...Finding a 280 Disassembler

It recently occurred to me, that with the introduction of the 280 upgrade for the SOL forthcoming from Micro Complex, there could be some possibility of converting 280 programs available for other machines for use on the SOL/280, and for those like myself who are running SOL/System B or similar machines under OUTER. Graphics programs may be a problem for the SOL and VDM users, but could be used by those using video boards with graphics capability. (SOL users could forego the main video circuitry and use an S100 graphics board such as the SSI 681.)

I have for some time been trying to get the Sorcerer version of "Adventure" running on my system—its cassette format is virtually identical to, and apparently copied from, CUMS. The only differences are the sixth character of the header (35 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 280 disassembler would help, but I can't find one. Perhaps some liaison with the SORCERER users group could produce some results—such as a short interface routines to enable cross-loading of CUMS and Sorcerer tapes. We would benefit by access to their

280 software and they could make use of the Proteus Cassette Library, particularly the Disk-tape program. Hope these ideas don't sound too far-fetched! Oh, and don't forget all that TSS-OE-system software waiting to be converted!

153 Twin Terraces
Edmonton, Alberta
CANADA, T6K 1V4
22 February 1982

Alastair Preston

...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 Ross's 80x24 screen mod. I haven't purchased it yet, but I probably just a matter of time (and money). I also think the 5-80 mod would be a real asset for our SOL machines.

I am still using a single density 8h disk controller and will probably upgrade to Bob Ross's new double density, no compatible controller board. However, I would like to hear from the other users who have purchased it, and find out how they like it.

A couple of months ago I implemented the 'key' reset hardware mod submitted by Jack Kinney, Vol. 4 no. 1, and it works great. It eliminates the need to hold down the reset 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 set his permission to pass it along to PROTEUS if other readers want to try unless I wish. It works well, however some of PR's early sense 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 set CP/M up and running under NH DOS. I set another SOL owner in town the other night at a CP/M users group meeting, and he has version 2.2. No reluctance so far has been the fact that I do not have a DE Dual disk system yet. I seem to think that CP/M needs a DD Dual disk system to be really effective.

Best Regards,

Rick Downs

P.O. BOX 1462 • AURORA, COLORADO 80012-1462

RICHARD E. DOWNS JR.
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 P. Otte

ALS-8 assembly language development system in ROM for SOL--$75. Extremely fast 8080 assembler, editor, simulator on 8K 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, 8-100 controller board (SD) and cable. Also a 16K RAM Memory board: $350.00 or best offer. Contact: Terry Walters, 441 Swallow Ct. Livermore, CA 94550 (415) 449-0566.

FOR SALE: SOL 64K with 2 slot Helios and Panasonic monitor/TV. Software includes Word Wizzard, Mail Master, Fortran (which has been debugged and works!), Regular Basic, Basic org. at D000 and Test Program on Disk and tape. Original Fortran, Basic and SOL System Manuals, PTOS manual zeroxed. Condition excellent with hardware updates where indicated. This, as my other two SOLs has always 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: Centronics model 700A, 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 Microplex 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 ES 232 with tractor.Has SOL Rev.0 with 44K memory, PTOS 1.5 (mod 2), Serial port driver for printer, Extended Disc basic and several games, etc. Excellent condition and full documentation.

N.W. Chamberlain, 1255 Hawthorne St. Alameda, CA 94501 (415) 521-0220

For Sale

Several SOLs available at Comart Limited, Little End Road, Eaton Socon, St Neots, Huntingdon, Cambridgeshire PE19 1JG. Telephone 0450-215005 Telex 32514 Comart U.

IBM model 2135 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 99509


WANTED

Documentation for a Wang model 701A printer.
Gary E. Lunsford Home (803) 766-0909
1688 Halo Lane Work (803) 554-5565 X 427
Charleston, SC 29407

A program for SOL-20 (32K Memory-Tape/Cassette Storage) RTTY to run on 8080.
Fred Salama W9WQ (415) 228-2817 Home-leave message on recorder (415) 645-0088 Office-leave message

PTC 3 P+5, 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 & Tlkin version of FORTH, for running on a CP/M 2.2 system.

Mick Kerwick
17 Chapel 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
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>McVIDEO UPGRADE AVAILABLE by Stan Sokolow</td>
<td>1</td>
</tr>
<tr>
<td>NEW VERSATILE DISK CONTROLLER by Stan Sokolow</td>
<td>1</td>
</tr>
<tr>
<td>65K RAM MEMORY MODIFICATION FOR THE SOL by Jim Spann</td>
<td>3</td>
</tr>
<tr>
<td>HARDWARE INTERRUPTS FOR THE SOL by Stan Sokolow</td>
<td>4</td>
</tr>
<tr>
<td>PER SCI 277 TO 270 CONVERSION by David Reis</td>
<td>5</td>
</tr>
<tr>
<td>LOCATING SAA 1027 CHIP AND CONVERTING EBCASIC TAPE TO DISK by John Whiting</td>
<td>6</td>
</tr>
<tr>
<td>PRODUCT REVIEW: EBCASIC TO DISK UNDER C/PM by TAD Enterprises</td>
<td>6</td>
</tr>
<tr>
<td>AN ADS NOISEMAKER FOR A 38 KEY PIANO by Frank Giziinski</td>
<td>7</td>
</tr>
<tr>
<td>PHONE NUMBER CORRECTION FOR DATA DELAY DEVICES</td>
<td>8</td>
</tr>
<tr>
<td>ENCYCLOPEDIA PROCESSOR TECHNICA IS READY</td>
<td>8</td>
</tr>
<tr>
<td>FIXING SOL TO &quot;VANISH&quot; by Wayne Wilson</td>
<td>9</td>
</tr>
<tr>
<td>IMPROVED ASL2 MULTISWITCH DRIVER by Wayne Wilson</td>
<td>9</td>
</tr>
<tr>
<td>REVIEW OF V10-X VIDEO BOARD by Wayne Wilson</td>
<td>10</td>
</tr>
<tr>
<td>ELECTRIC PENCIL USING N*DS by John Osader</td>
<td>11</td>
</tr>
<tr>
<td>24 X 80 CONVERSION AND THE NOVICE INSTALLER by P.N. Barnhart</td>
<td>12</td>
</tr>
<tr>
<td>HELP NEEDED INTERFACING N*DS 5.2D TO SOL by R.R. Walling</td>
<td>12</td>
</tr>
<tr>
<td>RING THE BELL ON A SERIAL DEVICE BEFORE PRINTING by D. Dalva</td>
<td>12</td>
</tr>
<tr>
<td>8D EXPANDOGRAM I MEMORY, 16KRA DELAY, ANOTHER ELECTRIC DRIVER by D.A. Howe</td>
<td>13</td>
</tr>
<tr>
<td>SOL-20 BACKPLANE PROBLEMS by Bob Marsh</td>
<td>13</td>
</tr>
<tr>
<td>H/2-25 PRINTER—EQUIPMENT REVIEW by Jeff Tom</td>
<td>14</td>
</tr>
<tr>
<td>MICROBYTE IN SAN JOSE HAS SOL PARTS FOR SALE</td>
<td>14</td>
</tr>
<tr>
<td>A SIDE BENEFIT TO ELECTRIC PENCIL VERSION SS by H.L. Winter</td>
<td>15</td>
</tr>
<tr>
<td>ADAPT ELECTRIC PENCIL TO SOL AT FOODHEX FROM CO0 by Bucklin</td>
<td>16</td>
</tr>
<tr>
<td>HELP NEEDED ON 1200 BAUD TAPES, G2 MICROSOFT AND MCKELVEY'S MCP by Alastair Preston</td>
<td>17</td>
</tr>
<tr>
<td>MORE ON MY GENERAL LEDGER PROGRAM by Franz J. Hirner</td>
<td>17</td>
</tr>
<tr>
<td>AN ALTERNATE B2X24 VIDEO UPGRADE by Michael Carter</td>
<td>17</td>
</tr>
<tr>
<td>RELOCATE N*DS AND INTERFACE EBCASIC TO A MICROPOLIS DISK DRIVE by Ron Shenk</td>
<td>18</td>
</tr>
<tr>
<td>THE AMATEUR'S GUIDE TO A 62K CP/M SYSTEM WITH MCOLOS AND DISC JOCKEY 2D, VERSION B by P.N. Barnhart</td>
<td>20</td>
</tr>
<tr>
<td>TELECOMMUNICATION SYSTEM FOR THE SOL/HELIOS II COMPUTER by Allen Fincher</td>
<td>22</td>
</tr>
<tr>
<td>KEYBOARD IN THE TROPICS by H.L. Winter</td>
<td>24</td>
</tr>
<tr>
<td>LETTERS TO THE EDITOR:</td>
<td></td>
</tr>
<tr>
<td>In Response to B. Diller's Keyboard Fix by Jeff Tom</td>
<td>24</td>
</tr>
<tr>
<td>on MCP/M, N*Disk, The Last Memory by R.C.F. Bartels</td>
<td>25</td>
</tr>
<tr>
<td>Finding a 280 Disassembler by A. Preston</td>
<td>26</td>
</tr>
<tr>
<td>Key Repeating and Underlining by R.E. Downs</td>
<td>26</td>
</tr>
<tr>
<td>UNCLASSIFIED ADS</td>
<td>27</td>
</tr>
</tbody>
</table>

---

**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.

And more, and more....
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 not really be obsolete, and you would continue to appreciate software being developed for it.

Solar owners should be quite familiar with the transistor 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 computer system to another dissimilar system.

So, standards were agreed upon for some of these 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.

CONT'N ON PAGE 2

WHAT'S NEW?

Datamation magazine July 82 issue has a big article on privacy in home information systems (interactive cable-television, telephone-based networks). It has some interesting facts and ideas: the FTC came up with 60 possible consumer services for such systems, in 6 categories: home banking, home investments, 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 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 Winchester diskette formats like an 8" or 5 1/4" floppy or 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) 297-5563.

Destek has announced a "starter kit" for its 510 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, broadband, 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 510 boards (one for each node), two coaxial cable taps, cable RG-59U coax), manual, and software (including source) on CP/M diskette ($5.95). 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.

Visual Group is the VisiCalc users group. They are also interested in users of other spreadsheet computer systems. Write to Visual Group, P.O. Box 254, Scarsdale, 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 "UNISON COMPUTER" put on the "US" festival in San Jose, California. Rock music, country western, homebrew computers,

CONT'N ON PAGE 18
So, the operating system should be considered just as important for portability as the language (or microprocessor) because the machines 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 even make any difference what kind of computer 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 just 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 computer to their students. But the project grew, and was licensed to Softech Microsystmes 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, or CP/M-86, or a P-System operating system as options has given IBM another boost. P-System, too. Before this, both Apple and NorthStar offered versions of the P-System, but now that IBM has given its endorsement, it seems more manufacturers are interested in it, too. I've noticed that Zenith, Texas Instruments, Philips, and others have announced their 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 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, an operation code 162 means "add integers", and it adds the top two integers in 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. For example, it is possible to run 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 some day, 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 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 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 II (6502 microprocessor), or with any 8800 or 280 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 Research 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.

If you purchase software that runs under CP/M and then you get a Z-8000 processor, you must say goodbye to all of your packaged software and buy new programs written in Z-8000 code. And what if your favorite microcomputer board is not supported either? Then 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: 8080/280, PDP-11/LSI-11, 8086/ 8088, 6502, 6809, 68090, 9900. Other vendors support p-code 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. Softetch sells compilers that trans late the following languages into p-code: UCSD Pascal, FORTH, 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.

Softetch 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..._page 3)
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 8088 with Heinos disk, and when it's ready, I'll make it available through Proteus. The Versatile Disk Controller (Proteus item M5) will be used in line. For other disk controllers, you'll have to adapt 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 (for 80186 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.

```
<table>
<thead>
<tr>
<th>High Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System Kernel (always resident)</td>
</tr>
<tr>
<td>BIOS</td>
</tr>
<tr>
<td>Codepool</td>
</tr>
<tr>
<td>Heap</td>
</tr>
<tr>
<td>BIOS &amp; Interpreter</td>
</tr>
<tr>
<td>Low Address</td>
</tr>
</tbody>
</table>
```

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 overwriting ("thrashing"), thus speeding up execution time.

Programs in the Codepool are dynamically relocated when necessary, to make room for incoming segments. Separate 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 known as multi-programming (multiple programs for one processor), but 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 single 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-storage devices, remote data transmission 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.
All I/O operations set a completion code (IORESULT) that can be tested automatically or by explicit instructions 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 doing 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 "ADDRESS" 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 ask for 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 CPM 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 books and Pascal compiler documentation, Softex sends the following books when you order the system:


Beginner's Guide for the UCSD Pascal System, by Kenneth L.


You may also find useful:

Introduction to Pascal (including UCSD Pascal), by Rodray Zaks, Sykes, 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 compiler 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 280, and for the 8086/8068.

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-code programs into 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 viewports ("windows") on the screen may be defined. Images may be in color or monochrome.

CON'T ON PAGE 5
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 (not 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-computers. 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 print request is added 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/Mellos.

Console Set-Up

The P-System provides video-oriented console I/O that is the same regardless of terminal type. That is, there are routines 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 functions. (That is, the control characters have different meanings for different 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 acceptance 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, IGAM (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 software 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.

5

CON'T ON PAGE 6
The prices may be different now, but prior to June 18, 1982, the adaptable p-SYSTEM cost $375. This includes the operating system, file handler, interpreter, two editors, p-code debugger, assembler for your host machine (both 8080 and 2-80 assemblers are included in the 8080/280 package), UCSD Pascal debugger, linker, utilities, and documentation package.

FORTRAN 77 compiler is also $375, BASIC is $225; a package of assemblers for various processors (280, 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 8080/280 native code generator is $150; the print spooler is $50; Xenofile (CP/M Interchange utility) is $50; Turtle Graphics 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.

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

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

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

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

6. 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 HTYPE 1 PRINTER DRIVER
by Dean Mazur

Processor Technology committed a few blunders with their printer driver which has been distributed with their system. 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. New labels and changed abbreviations have been made that those with access to the PT manual for the interface card may easily compare this driver with their version.

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.

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

The entry point assumes the character is in register C, as per CP/M, and not in register B. Also, the formfeed character is property defined as DCH, not D0H.

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 Htype can take several seconds, depending on the position of the carriage when the check condition occurred. The W3 timer usually expires before the restore complete, 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 SOUL equate) and set the error flag. Then, on the next call to the driver, a restore sequence is forced, but not at top margin, and ALIM FORMS is sent to prevent an operator from 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).

Because of the size of the driver, we loaded 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 line input, and 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.)

The PT Htype controller card is a very simple, reliable unit and appears to be well designed. If you use 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 different works of the printer (except software control of ribbon lift for two color printing).

Finally, the user is free to alter margin values, deltas (used to define Elite or Pica pitch) and options. Instead of you write a short, separate program to configure these interactively so that you don't have to fool around with D0T or D0H 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 Alton Mazur
505 W. 8th St.
Port Angeles, WA 98362

CONT'N ON PAGE 7
CON'T FROM PAGE 6

2708 EPROM BURNER PROGRAM

**TITLE '2708 EPROM BURNER PROGRAM'**

0100 ORG 100H

0000 = WBOOT EUQ 0003H ;WARM BOOT POINT
0003 = BBS EUQ 0006H ;BBS ENTRY POINT
0006 = CONOUT EUQ 0002H ;CONSOLE OUTPUT FUNCTION
0009 = PSTRING EUQ 0001H ;PRINT STRING FUNCTION
000AH = READBUF EUQ 0000H ;READ CONSOLE BUFFER FUNCTION
000DH = PROM EUQ 0008H ;BASE OF PROM IN USER SYSTEM
000FH = LQ EUQ 0007H ;CARRIAGE RETURN
0010H = LF EUQ 0006H ;LINE FEED

LOAD SP AND PUT OUT SIGN-ON MESSAGE

0100 51C002 
0103 1FFA01 
0106 C6EF02 CALL PRINT ;PRINT MESSAGE AFTER CR/LF

GET SOURCE ADDRESS AND PERFORM HEX CONVERSION

0109 111902 SOURCE: LXI D,DPROMPT1 ;ENTER SOURCE ADDR PROMPT
010C C6EF01 CALL PRINT ;PRINT SOURCE ADDRESS

010D 060A MOV C,READBUF ;READ BUFFER FUNCTION
0111 119402 LXI D,CONOUT ;POINT TO THE BUFFER
0114 C00500 CALL BBS ;GET INPUT LENGTH
0117 5A9502 LOA CUNSIZE ;GET INPUT LENGTH
011A 8E0D CALL 5

012C 1CD001 CZ BADDR ;COMPLAIN IF NO INPUT
012F C6C001 CALL SOURCE

012A 219602 LXI H,CONLIN ;POINT TO LINE PORTION
012B 010000 LXI B,0 ;CLEAR BC
0130 4A MOV C,A ;PUT SIZE IN LOW ORDER BYTE
0131 09 DAD B ;SLIDE POINTER TO END + 1
0132 3624 MWI M,5'8' ;MARK IT TO KEEP CP/M HAPPY
0134 119402 LXI D,CONLIN ;POINT TO LINE PORTION AGAIN
0137 210000 LXI H,0 ;CLEAR HL

013A 1A HCONV: LOA X ;GET CHAR FROM LINE PORTION
013B C6F24 CPU 5 ;CHECK FOR END OF LINE
013C C53001 JZ TARGET
0140 20 DAD H
0141 29 DAD H
0142 29 DAD H
0143 29 DAD H
0144 C0C001 CALL HEX
0147 9E6001 CNC BADDR
014A 020001 JNC SOURCE ;TRY AGAIN

014D 85 ADD L
014E 6F MOV L,A ;MOVE IT IN
014F 15 LJE B ;JUMP POINTER

0150 33A01 JMP HCONV

GET TARGET PROM NUMBER AND CONVERT IT

0153 229002 TARGET: SHLD SAADR ;SAVE SOURCE ADDRESS
0156 113502 TARGET: LXI D,PROMPT2 ;ENTER TARGET PROMPT

0159 C0EF01 CALL PRINT
015C 060A MOV C,READBUF
015E 119402 LXI D,CONOUT
0161 00500 CALL BBS
0164 349502 LOA CONSIZE ;GET INPUT SIZE

0167 F001 CPI 1
0169 C4D1 CHZ BADRIP
016C 255601 JNZ TARGET
016F 119602 LXI D,CONLIN ;POINT TO LINE PORTION
0172 1A LOAD D ;GET CHAR
0173 F300 CPI 0 ;IS IT A 0 ?
0175 DC00B1 CC BADRIP
0176 DA5601 JC TARGET
0178 F658 CPI 4 ;IS IT AN 8 ?
017A D45001 CNC BADRIP
017D 255601 JNC TARGET

0180 025601 SU3 30H ;BETWEEN 0 & 7 SO REMOVE BIAS
0183 0230 NOV B,A ;SAVE PROM NO. IN B

CONVERT PROM NUMBER TO A TARGET ADDRESS

0186 110004 LXI D,124 ;SIZE OF A 2708
0189 0500E0 LXI H,0 ;POINT TO BASE OF PROM
018C 97 SUB A ;0 INTO A
018D 00 LOOP: CMP B ;COMPARE TO PROM NO.
0191 CE3901 JJ BURN ;JUMP WITH CONVERSION

0193 C96001 INR A ;ADD TO CONVERSION ADDRESS
0195 1C JMP LOOP ;BUMP COUNT
0199 17 ADD B,A ;ADD 1/264 TO BASE ADDRESS

NOW WE CAN BURN THE PROM

019C 22E002 HURN: SHLD TADDR ;SAVE THE TARGET ADDRESS
019F 110004 LXI D,124 ;SIZE OF A PROM
019C 19 MOV D,A ;GET HIGH ORDER BYTE
019D 7C MOV A,0 ;GET HIGH ORDER BYTE
019E D9 DIPROM ;SAVE IT FOR LATER
01A1 010000 LXI B,0 ;PASS COUNT = 0
01A4 28E002 HLOOP: LHLD TADDR ;GET TARGET ADDRESS
01A7 7A XCHG
01A8 2A9302 LHLD SADDR ;SOURCE ADDRESS
01AB 7A HLOOP: MOV A,D ;GET COUNTER VALUE
01AC 35 POP H ;SAVE IT FOR LATER
01AD 219002 LXI H,TSTVAL ;POINT TO TEST VALUE
01AE 00 MOV C,M ;COMPARE WITH TEST VALUE
01B0 0E POP H ;RECOVER TARGET POINTER
01B2 C8C01 JZ NEXT ;NEXT PASS IF IT MATCHES
01B3 5E MOV A,M ;GET DATA BYTE
01B5 1E ADD X ;ZAP THE PROM
01B7 23 INX H ;BUMP SOURCE POINTER
01B9 0E INX D ;BUMP TARGET POINTER
01BA 09 JMP LOOP ;CONTINUE
01BC 03 NEXT: INX B ;BUMP PASS COUNT
01BD 78 MOV A,B ;
01BE F004 CPU 1 ;1K PASSES YET?
01C0 C2A401 JNZ LOOP ;KEEP GOING IF NOT

TERMINATION MESSAGE

01C3 117102 LXI D,TERMMSG ;SAY BYE-BYE
01C6 C5EF01 CALL PRINT ;JMP WBOOT ;BACK TO CP/M

UTILITY SUBROUTINES

01CC 0630 HEXIT: SU1 4B ;REMOVE ASCII BIAS
01CD 06 FA CPI 10
01D0 90 RC
01D1 0607 SU7 7
01D2 03 F0 CPI 10 ;IF LESS THAN 9
01D5 97 RET ;IT'S A LETTER
01D6 0E02 PUTCHR: MWI C,CONOUT ;CONSOLE OUTPUT FUNCTION
01D8 05 MOV D,A ;SCHARK TO SEND IN RIGHT PLACE
01DA 090050 CALL BBS
01DC 59 RET
01DF 115302 BADRIP: LXI D,ERRMSG ;POINT TO ERROR MESSAGE

CON'T ON PAGE 8
CON'T FROM PAGE 7

0130 C6EF01 CALL PRINT
0133 3E0 CALL LW A,6 ;CARRIAGE RETURN
0136 C6EF01 CALL PUTCHR
0139 3E0 CALL LW A,6 ;LINE FEED
0142 3E0 CALL PUTCHR
0145 C6EF01 RET
0148 0F5 PRINT: PUSH 0
014B 3E0 CALL CRLF
014E 01 POP 0
0151 4808 PRINT CSTRING ;PRINT STRING FUNCTION
0154 3E0 CALL BB0S
0157 C6EF01 RET

; STRING DATA AREA FOR CONSOLE MESSAGES

015A 3237301280 DESIGN: DB '7/26 PROM BURNER - Version 1.08'
0163 456745572P@PMT1:DB 'Enter source address (hex): $'
0166 456745572P@PMT2:DB 'Enter target PROM (U = 7): $'
0169 4565245260@PMT1:DB 'ERROR: Incorrect input format
016C 427572620E38MM:DB 'Burn completed...returning to CP/M$'

; VARIABLE AREA

0294 05 CONBUF: DB 5 ;LENGTH OF BUFFER
0297 CONSIZE: DB 1 ;SIZE AFTER SUCCESSFUL READ
029A CONLIN: DB 0 ;BUFFER PLUS 1 MORE FOR $0
029D SADRM: DB 2 ;SOURCE ADDRESS
029F TESTVAL: DB 1 ;TEST VALUE (EQUALS 0 FOR FCD SOURCE)
02A2 TADDR: DB 2 ;TARGET ADDRESS IN 27UB
02A5 STACK: DB 32 ;STACK LEVEL STACK
02B0 STACK: DB 25 ;STACK END

ED. NOTE: Modified Nytype 1 Printer Driver program added to
EPT Volume II.

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-29, 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 "X"
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="P820"
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: SET OUTPUT FILE TO VDM DRIVER AND LINE LENGTH TO 64
210 REM
220 SET OF=80
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 CONFIGURATIONS": 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=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 620
460 GOSUB 620
470 PRINT TAB(7):"PORT I --- I PERIPHERAL I;"
470 PRINT TAB(50):"FUNCTION;"TAB(7);"I HEX"
480 PRINT TAB(4):"CONNECTION I --- I CONNECTION I;"
490 PRINT TAB(7);"I MASK"
500 GOSUB 690
510 REM
520 REM: PRINT FORM LINE
530 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 etc., 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 1R5
14 July 1982

Yours truly,

Alastair Preston.

(Ed. note to Alastair Preston:
The bulletin board systems are great, but they don't seem appropriate 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... the 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.

Sincerely yours,

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

---

SUPER-PHANTOM FOR SOL-20

by Father Thomas McGahee

December, 1981

LETS YOUR SOL-20 HAVE A FULL 64K OF RAM AND SOLOS AT $0000

SUPER-PHANTOM FOR SOL-20

by Father Thomas McGahee

December, 1981

LETS YOUR SOL-20 HAVE A FULL 64K OF RAM AND SOLOS AT $0000

ULD-H-20 (OP)

7474 or 74-LS74 "D" FLIP FLOP

ANY GERMANIUM DIODE

IN 270 ELECTS

U34-3 (memory decoder)

TO ALLOW RAM TO OVERLAY
RC000H - DCFFFH,
MVI A-01
OUT OFFH

TO RE-ACTIVATE SOLOS,
MVI A-00
OUT OFFH

8T97, 8097 or 74-367

PHANTOM

J1-67 (5-100 BUS)

PHANTOMX (NEW PHANTOM)

J1 or (U77-9)

MWRITE

U24-13

or (U50-li)

MEM. SEL.

U33-6

U24 74LS04

MEM. SEL.

U33-6

PHANTOMX occurs during POC, and also whenever SOLOS has been activated and an address from 0000H - DCFFFH is on the address bus. Whenever SOLOS is active, MWRITE will be LOW (disabling memory writes to plug-in memory). Together, PHANTOMX and MWRITE allow RAM and SOLOS to peacefully coexist.

CON'T ON PAGE 10
Dear SOL Brothers,

In the May/June/July/August 1981 issue of PROFILES (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 an all-out "tickle" story, but it isn't something you should miss!

Here's a method I present a method that overcomes the problems associated with the Spann/Athey method. This modification is currently running on my 74LS74, 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 SOLBUS, RAM and VIDEO RAM. It is totally compatible with all previous hardware/software, so 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 CMOS-CFFTH.

The Spann/Athey modification involves tricking the memory decoder into ignoring addresses from CMOS-CFFTH 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 CMOS-CFFTH 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 READIN. WRITING 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 all.

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 WRITING 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 4744) is connected as a one-bit port. The DS pin is disabled by pulling it to +5, and the DS pin is connected to PC6 (Power-On Clear). The DATA pin is connected to the D8 line, and the CLOCK pin is connected to the FF output of the Output Port Decoder. This is a very good reason for choosing the FF port instead of the I/O port as suggested by Stan. If the FF port were used, then sending 00 or 01 out to that port could inadvertently do such bad things as turn off the tape recorder.

For years, I somehow just didn't like this at all. However, if you put FF port sidesteps all the possible hidden problems that might otherwise arise. I tend to be the cautious type. I used port 16H originally for both, but have since changed my mind since I am avoiding any possible conflicts with Bob Hogg's 8X24 modification board for SOLs. The potential problem here is that he uses port FF for the DPLAY function. You don't really want to be including or deactivating or de-activated by sending out a BELL code, now do you?)

One of Hogg's 8X24 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 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 using 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 pin 7 and 14 together. When 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 all wires 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 them into this socket, and solder all wires to their proper I/Os. I removed the IC in question, turn it upside-down, and tin the SHOULDER of the proper pin. I then spot solder the proper with wire in place. Keeping the IC upside-down helps keep the solder on the shoulder.

There is an empty socket at location U183. 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 insert the 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, we can get close to activating PHANTOM inside the 74LS74 lines. This forms OFF-BOARD memory that READING and WRITING are not allowed. Whenever MEM SEL is LOW, indicating that SOLOS is inactive, then the normal PHANTOM and WRITING signals are put on the bus via tri-state buffers A and D. Because of the PHANTOM and WRITING signals, off-board RAM/RAM can coexist with SOLOS RAM/RAM. Which one will respond to the addresses from CMOS-CFFTH is determined by the one-bit port that we installed using the 74LS74.

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

**MVI A81 OUT 0FFH**

**MVI A8B OUT 0FFH**

I used a 74367 as the tri-state buffer, and soldered it the empty position marked [U82] on the SOL PC Board. (An alternate position is U83, or piggy-back it onto an 8779.) In any case, pin 8 is ground, and pin 16 is 5 volts. Please take note of the following:

- All ON-BOARD PHANTOM and WRITING SIGNALS ARE LEFT INTACT. ONLY THE BUS SIGNALS BECOME PHANTOM AND WRITING. If you have an empty position, then solder in a socket to add 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 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 74367 to ground (pin 0).

Next, isolate the on-board PHANTOM line from the J1 bus. Using page X-3, locate pins 67 of J11. Notice that it connects to a feedthrough 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 feedthrough to pin 2 of our 74367. Daley-chip pins 13 and 7 and 13 of our 74367 to pin 67 of J11 (bottom of board). Next to pin 67 of J11 is pin 69. On the bottom of the board, cut the run that goes to this pin, and connect pin 69 to pins 3 and 11 of U22 we did 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 feedthrough next to R28 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 U83 position).

**CON'T ON PAGE 11**
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 74LS04 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, 01 / OUT 0FFH at the very beginning of the boot program on DISK. This requires 4 bytes. Some boot loaders may not have enough room. Remember, we're only really concerned about getting 88 high, so if the boot routine loads any register with data that has 88 high, then use a register exchange operation to get it copied into "A", and then output it to port 0FFH. 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 SOLOFF and SOLOW routine like these:

SOLOF EQU FF00h ;SOLOF = NAME OF PORT.
CNX EQU 05h ;05h = TURN SOLOF ON.
OFFX EQU 01h ;01h = TURN SOLOF OFF.

SOLOW: PUSH PSW ;SAVE "A".
MVI A, OFHX ;0h MEANS TURN ON SOLOF.
OUT SOLOF ;DO IT.
POP PSW ;RECOVER "A".
RET

SOLOFF: PUSH PSW ;SAVE "A".
MVI A, 0FFH ;01h MEANS TURN OFF SOLOF.
OUT SOLOF ;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 SOLOF :ACTIVATE SOLOS.
CALL SINP :CALL SOLOS SINP.
CALL SOLOF :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, SOAT, SOINT, SSTAT, etc.

**POINTS TO CONSIDER**

In a CP/M system, the BIOS may NOT be in the range C800H-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 CP/M the STACK may not reside from C800H-CFFFH, since the stack is used for CALLs and RETs. Should you find this to be a problem in your system, it can be easily cured by patching SOLOW and SOLOF to include a single stack swap routine. SOLOW would switch to a temporary stack (it need only be about 24 bytes deep), and SOLOF 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:

SOLOW EQU FF0Fh ;SOLOW = NAME OF PORT.
CNX EQU 05h ;05h = TURN SOLOF ON.
OFFX EQU 01h ;01h = TURN SOLOF OFF.

SOLOW: PUSH PSHW ;SAVE "A".
MVI A, 0FFH ;01h MEANS TURN ON SOLOF.
OUT SOLOF ;DO IT.
POP PSHW ;RECOVER "A".
SHLD TEMPH ;... GET RETURN ADDRESS.
SHLD RETX ;... AND STORE IT.
SHLD SPX ;0L = STACK-POINTER.
SHLD OSTACK ;SAVE OLD STACK ADDRESS.
SHLD SP ;SET UP TEMPORARY STACK.
SHLD RETX ;RECOVER RETURN ADDRESS.
SHLD SP ;PUT IT ON STACK.
SHLD TEMPH ;RECOVER ORIGINAL HL.
RET

SOLOFF: PUSH PSHW ;SAVE "A".
MVI A, 0FFH ;01h MEANS TURN OFF SOLOF.
OUT SOLOF ;DO IT.
POP PSHW ;RECOVER "A".
SHLD TEMPH ;... GET RETURN ADDRESS.
SHLD SPX ;0L = STACK-POINTER.
SHLD OSTACK ;RECOVER OLD STACK ADDRESS.
SHLD SP ;MAKE IT CURRENT.
SHLD RETX ;RECOVER RETURN ADDRESS.
SHLD SP ;PUT IT ON STACK.
SHLD TEMPH ;RECOVER ORIGINAL HL.
RET

TYPH DW 00h ;HL STORAGE.
RETX DW 00h ;RETURN ADDRESS.
OSTACK DW 00h ;OLD STACK ADDRESS.
LSTACK DS 24 ;LENGTH OF STACK IS 24 BYTES.
TSTACK DB 00h ;[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 C800H. 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 BIOS (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 1800H to 1830H, which is a total of 23 records. (16 x 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 1660H. Then at 1800H I patched in the following:

CON'T ON PAGE 12
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, 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 one 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. If 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 compatibility 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.

The 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 Nog. Incidentally, I urge you to use his Dual Personality module for replacing the SOLOS at C00 (this is the original 64x16 SOLOS), and Bob Nog's 24x80 enhanced SOLOS monitor ALSO AT C000. He makes many versions of this monitor... all at the same cost. But you must be explicit in telling him what versions you want! In my system I put the 24x80 enhanced SOLOS into the socket labelled "F000." Since it is actually at C000, I had to make a minor change to his instructions that came with the 88x24 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 F000 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 any of my systems, my disk controller resides at F000, making a F000 system imposssible. With a C000 system, the only thing you have to make sure of is that the portion of your BIOS that the SOLOS I/O are above CFFF. If you have a disk controller sitting at an inconvenient location, you can always go with a F000 system. In this case, you can get enough out of the overlay RAM by loading utilities at F000 and then relocating them from F000 up. Just make sure such utilities don't try to directly access the monitor routines or they will crash. By the time you get the basic idea, carefully look over your system needs before deciding on a C000 or F000 system.
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 I DUAL DENSITY CONTROLLER IS A WRITE STAR LOCK - LIKE CONTROLLER BOARD. IT IS PROVIDED WITH IT'S OWN DOS CALLED "MICROS". THE DOS IS ALSO COMPATIBLE WITH EXISTING NORTH STAR SOFTWARE. HOWEVER ITS INTERNAL STRUCTURE IS CONSIDERABLY DIFFERENT. SOME OF THE FEATURES THAT MAKE THIS BOARD AND DOS AN EXCELLENT CHOICE ARE:

1. WITH AN OPTIONAL BOOT ROM AND DIP SWITCH THE BOARD IS RELOCATABLE.

2. IT WILL WRITE USING THE NEW TANDON DOUBLE SIDED DRIVES IN BOTH THE 48STI AND 90STI. THIS ALLOWS US TO DISK BYTES PER 80STI DRIVE - INCREDIBLE!

3. THE DOS DOES NOT USE ANY MEMORY ABOVE 256, SO ALL NORTH STAR SOFTWARE WILL WORK - EVEN PROGRAMS THAT START AT 256.

4. THE I/O AREA IN MICROS COMES CONFIGURED FOR A "STANDARD" SOL AT GOOD (LOAD & GO).

AFTER PURCHASING THE CONTROLLER I DECIDED THAT A SECOND DRIVE WAS NEEDED ESPECIALLY IF I WAS GOING TO GET CP/M UP, SO I BOUGHT A TANDON NEW 256. THIS IS A 48STI DRIVE WITH DUAL HEADS SO IT WRITES ON BOTH SIDES OF THE DISKETTE. WITH DOUBLE DENSITY FORMATTING THIS ALLOWS YOU TO PUT ABOUT 4096 BYTES ON A SINGLE 5 1/4" DISKETTE - THAT'S A LOT OF STORAGE! THE BOARD AND DISK HAVE BEEN OPERATING FLAWLESSLY FOR OVER A MONTH - I AM VERY PLEASED WITH MY NEW SYSTEM'S 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 SOLIDS UP TO 600 FOR USE WITH CP/M, BUT IT ALSO HAS AN ENHANCED VERSION OF SOLIDS (MICROS) 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 FOLLOW THAT ARE AVAILABLE WITH THE NEW "MICROS".

DA = DISPLAY ASCII MEMORY DUMP
DI = DISPLAY HEX MEMORY dump
EA = ENTER ASCII CHARACTER STRING
EH = ENTER HEX BITE STRING
FM = FILL MEMORY
MM = MOVE MEMORY BLOCK
CM = COMPARE MEMORY BLOCK
AS = ADDRESS REFERENCE LOCATOR
FH = FIND HEX WORD OCCURRENCES
FX = FIND HEX BITE OCCURRENCES
FA = FIND ASCII CHARACTER
FC = FIND ASCII CHARACTER SINGLE
TM = TEST MEMORY

ALSO PROVIDED ARE SOME JUMP VECTORS THAT CAN BE MODIFIED WEEK ORDERING FOR A MINIMAL CHARGE. THESE ARE -

CS = VECTOR TO DISK OPERATING SYSTEM @ 2026
WE = WARM BOOT VECTORS TO GOOD FOR CP/M
WP = WORD PROCESSOR VECTOR @ 202C
BA = BASIC RE-ENTRY VECTOR @ 2A4A
BC = BASIC RE-ENTRY V/CLEAR @ 2800
BV = BASIC RE-ENTRY NO RESETS @ 2A4A

ANOTHER NICE FEATURE IS THAT THE "LOAD" KEY ON THE SOL WILL NOW BOOT NORTH STAR AT 256. THAT VECTOR ADDRESS CAN ALSO BE MODIFIED WHEN ORDERING FOR A MINIMAL CHARGE. THIS MAKES IT A SNAP TO BOOT MY SYSTEM UP IN MICROS.

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 RAVEN'T CONVERTED TO CP/M SO THAT YOU CAN RUN OUT AND PLUNK DOWN $50 TO MICROSOFT FOR "WORDSTAR". SINCE THERE SEEM TO BE SO MANY MODIFICATIONS TO PENCIL - WHY NOT TRY AND OBTAIN THE SOURCE CODE FROM MICHAEL SOYER. IF THOSE OF US THAT WOULD LIKE TO KEEP E.P. ALIVE WOULD EACH KICK IN SAY $50.00 WE PROBABLY JUST MIGHT BE ABLE TO PURCHASE IT. ANY THOUGHTS??? THE REASON I MENTIONED THIS WAS BECAUSE AFTER I HAD MY NEW PHASE LOCK I CONTROLLER RUNNING I TRIED TO LOAD MY DISK VERSION OF PENCIL AND YOUGUESSED IT, IT DIDN'T RUN. THE DUAL DENSITY VERSION OF E.P. DID NOT USE DOS (SORRY WRITE HIS OWN DOS). IT ALSO LOADS AT 256.

SO NOW I HAVE BOTH MY OLD SINGLE DENSITY CONTROLLER, AND THE PHASE LOCK I CONTROLLER IN MY SYSTEM ADDRESSED AT DIFFERENT LOCATIONS - JUST SO I CAN RUN PENCIL. ANY IDEAS?????

I DID IMPLEMENT THE PENCIL MODIFICATION BY LEO WINTER, AND ENJOY THE INVERSE VIDEO ALLOT - THANKS LEO.

WELL, MY CP/M ARRIVES 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(7);C;
570 PRINT TAB(15);";TAB(22);";TAB(35);";TAB(71);";* 
580 GOSUB 690
590 NEXT C
600 PRINT
610 RETURN
620 REM
630 REM: PRINT BIANK 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 ABA language features to Pascal, and a macro preprocessor. The whole package costs $540, 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) 540-5581.
I agree with Lew Mosley. One of the major advantages of a SOL is the ability to exchange software between a large variety of 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 COOH.

(Ed note to John Barber: You can order the MCDM with standard Solos in the COOH 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
21 Feb. 1982

Dear Stan,

Has anyone come up with some method of cataloguing Helios Disks, somewhat in the same manner as can be done for CIM, using a line 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 layout about the 80 x 24 conversion in vol. 1 # 5 & 6 of PROTEUS is good news. But there may be some drawbacks, namely, does it prejudice the use of any disk drive that has a ROM in the F000 block? Moreover, can one use PTOS?

As many of our members, I too believe PTOS is a better operating system than anything else around, in fact makes CIM look positively cumbersome. Too bad that some who have control of the source code do not make use of it to make PTOS machine independent, so that programs could be written using it, or the same code, in fact, of course, we all know that the dictates of the marketplace have something to do with this, i.e. there is any money in it? Given the proper marketing that too could be overcome.

Witness the fine software that the Basic Computer Group in Vancouver wrote for WIZARD or MAILMASTER. A word processor such as WORD WIZARD is hard to beat, and I understand that GERALD LEKLIN has software control to get around PTOS 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 Alabamco Cr.,
Baltimore, Md. 21204

(Ed. note to Frank Sanders: Good news, A PTOS version of Greenlaw's cataloging system is in the Helios library for future releases. The 24 x 80 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 F000 instead of F000.)

---

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-62 on MURB 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 compatible format, i.e., North, or Diskus. It is difficult to get software in VISTA format. TOSFEDISK 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 60029, is a true conversion. It comes with a cassette file to which ECB is appended, which modifies 1/0 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 TExt 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 TExt format. Since there are two very easy ways to program around this it is no serious defect. I can load ECB from tape, load the file saved in semi-compiled format, and save in TExt format, or add Byte by Byte ECB to PROTEUS. The installation manual says that one can also use the conversion for Basic/S. 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 never did use Basic/S 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 compatible 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 card, 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 5-100 MICROSYSTEMS (Jan/Feb 1982) for an article on making SOL vanish. The trick seems to be to put SOLDIS and system RAM on the internal bus, rather than the S100 bus 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 FM seems easier to use.

CONTINUED...
...Booting Up the Sol with Northstar

TO DISCOVER the type of information you most desire we 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 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 8, 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/Micros users.

The software is a mini-wordprocessor for the Sol. Two versions are available: one disk-based using Micropolis' Mod II disk I/O (all other routines in MOS are used) and one tape-based using SOL55 tape routines. Both are offered for $45 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 on line at a time and 15 lines at a time; (4) Smart mode; (5) Delisions using Del key; (6) Black delete; (7) Sort routines; (8) Clear text buffer with one key; (9) "Home" command displays beginning of text before; (10) Paused 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 line length, left margin indication, and line spacing are controlled from the keyboard; (14) Printer commands may be embedded within the text; (15) Error feed; (16) Page numbering can be toggled on or off.

Although the printer driver was programmed for the Epson MX-90, 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-0, which are used to configure the printer.

At this point I would like to modify SWORD for use on other systems using MOS, but I need more information 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

...Join the Nevada Cobol Users Group

Nevada Cobol Users Group
5356 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 among users developing extensions 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.

Yours,

Douglas W. Stone
14th Whippornell
Oakland, Texas 75040
214 840 8534

- OR WORK -
STCR INC. AIRLINES SERVICES
1525 Elm St., Suite 2660
Dallas, Texas 75201
214 749 1983

Dennis C. Fait

15
... 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 am now 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 6K but have only enough chips for 4K.
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 05-11, 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 already 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 as on 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 McGehees provide a back spasing DEL function and a Diablo driver at 1200 baud. Some information was pooled 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. McGehee 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
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 the 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 Dijmbos or their kin, it also takes 3 times as long to print. Approx. 10 c.p.m. For those of you used to a dot matrix that zips along at 30-50 c.p.m., it will seem very slow. For me, I have SOL set at 110 baud, and although the TP-1 is adjustable, found that couldn't handle anything faster.

That is the only real problem I have found with it so far, and it took up several hours (straight) of time on some paper. It doesn't seem to overheat or long runs, and just keeps plugging along.

Rich Oshlack is looking for letter quality, and can't afford an expensive printer, and really mind the printer taking about the same amount of time as a feet human typeset, this is a way to go. Ann at Micro-printer Marketing is keeping me posted when other type styles are available, 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 expire, could you let me know? I want to keep it current)

Paul W. Kittie
PO Box 1286
Loma Linda, Ca. 92354

...HELP WANTED FOR REPAIRING SOLS by Ron Computing Systems
RON COMPUTING SYSTEMS INC.
362 Pineview 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 anywhere 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,
Galen D. Miller
Service Manager
... A Recommendation for Stetson Electronics

John A. Whiting
130 S. Coronado St. #31
Los Angeles, California
(213) 680-1912 1200 hrs. 90057

Greetings, and Salutations:

My name is A. Whiting, and I'm the one who wrote the previous letter. I just got the latest issue of PROTEUS NEWS and the first thing I noticed was how well your printer reproduced the issue. Your head is now just a bit larger.

Anyway, I'm writing this time to announce that I did find a source of the SAA 1027 chip, and the KIM 27 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; 03064, (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 driver kits, 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 patiently while I scrounged 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 shipping at all. Bob's price was $10.95, including the shipping.

Now I have to do is get the 98%!!! drive working. I heartily recommend Stetson Electronics. He doesn't promise more than he can deliver, but he manages to deliver twice as well.

For PROTEUS NEWS, it was a great issue. I won't mind it becoming a quarterly as long as the subsequent issues are as good as this one. Keep up the good work, folks!

Regards,

John A. Whiting
JAV/wol

Post Script: I enclose Bob's flyer in the hopes that it proves reproduable, and useful.

STETSON ELECTRONICS
P.O. BOX 3008
NASHUA, N.H. 03064
(603) 880-4975

STETSON ELECTRONICS HAS A WIDE VARIETY OF ELECTRONIC COMPONENTS AND ASSEMBLIES. IF IT'S NOT ON THE LIST, WRITE AND I WILL TRY TO ACQUIRE ONE FOR YOU AT A REDUCED COST.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>PRICE</th>
<th>SHIPPING</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 1/4&quot; DISK DRIVE ENCLOSED WITH SUPPLY ........ $65.00 EACH</td>
<td>$5.00</td>
<td></td>
</tr>
<tr>
<td>5 1/4&quot; DISK DRIVE WITH ENCLOSED AND POWER SUPPLY $175.00</td>
<td>$10.00</td>
<td></td>
</tr>
<tr>
<td>5 1/4&quot; DISK DRIVE NO ENCLOSED / NO SUPPLY ................ $125.00</td>
<td>$5.00</td>
<td></td>
</tr>
<tr>
<td>5 1/4&quot; DISK DRIVE SIGNAL CABLES (3A PIN) W / RIBBON CONNECTORS FITS ANY 5 1/4&quot; DISK DRIVE (SEE BELOW)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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-Z80 (SPECIFY VHTCH)
TRS-80 IS TRADE MARK OF TRS-80 CORPORATION
PLEASE SPECIFY LEVEL III OR LEVEL IIIA / MOD 1 OR MOD 2

TAPE OR DISK SOFTWARE AVAILABLE FOR APPLE
APPLE IS TRADE MARK OF APPLE COMPUTER CORP.
PLEASE SPECIFY TAPE OR DISK

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 8:30PM 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 CB IOS KEYBOARD INPUT PROBLEM

After reading Leonard's fix for the Vista CB IOS in Vol. 4, Issues 9/6, I was reminded of a similar problem I had with my 5-100 system. I also had problems with double, and even triple, characters being displayed whenever I typed anything. It turned out 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 CB IOS can get to the CIMP 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 CIMP fix shown in the article could be written as follows:

CIMP: CALL SINC ;Check keyboard
JZ SINC ;There's nothing there
ANI 7FH ;strip bit 7

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 Fair'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. Unison Corporation, 2001 Gateway Place, Suite 500, San Jose, CA 95110, (408) 294-8424.


CompuCover makes cloth-backed vinyl dust covers for almost every kind of computer and peripheral, even the Sol (514.95). CompuCover, P.O. Box 324, Mary Esther, FL 32569, (904) 243-5793.

CMRFPOWER 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).

---

CONNECTING THE EPROM MX-80 TO THE SOL by R. Shulkin

SIVAD of TEXAS

Roger Shulkin, Distributor

20214 Brionesbury Drive
Katy, Texas 77450

Texas 800 392-2452

(713) 492-1931

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 OLT 253.-

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:

1. On
2. Off
3. On
4. Off
5. On
6. Off
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:

<table>
<thead>
<tr>
<th>SOL</th>
<th>to Epson MX-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

I use what I know as the Serial Driver for TI-810, Centronics 761, and 108 123, 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, 0 \ 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 MoviDe video 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 2400) to release 5.2 (located at 0600). Any comments would be greatly appreciated.

Again, many thanks for your excellent articles and valuable information. Please continue!

Roger Shulkin

(End note to Roger Shulkin: I believe that all you need to do to make your existing NorthStar DOS talk to the 24x80 mode of MoviDe Video is just to change the User Area's CALL instructions which use the entry points into Solos. Instead of calling C006, you call 0004. When the video board is switched to 24x80 using the Dual Personality Module, the Sol and Solos go to the F Switch, 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.)
WANTED
ALS-8 assembly language development system in ROM for Sol. 8080 assembler, editor, simulator on 8K EPROM board by Processor Tech. Also wanted are the user documents and the 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 Sol17?
M.J. Kerwick, 17 Chapel St., Carrick-On-Suir, Co.Tipperary, IRELAND.

WANTED
Electric Pencil cassette 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 Techniques. 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 300/8 KSR terminals, with Tytype I mechanism and keyboard, 30 CPM, 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 address. SDRAM disable: 4 MHz
$110
DATA-SHIFT 24K STATIC RAM -- S-100 -- 4 MHz
$160
NORTHSTAR 25K 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.
$25
2 back plane boards for the Sol-20
$6 each
270B Personality module board
$12
30A1A-W Hideout for the Sol -- set of 2
$10
SOFTWARE TECHNOLOGY music suite
$20
CASSETTE RECORDER -- Tape counter, cords, auto-level control
$20
EXTENDED CASSETTE BASIC
$25
TREK-90 & VARIOUS GAMES
$10
EXTENDED DISK FORMAT MANUAL
$3

All items work and are in good condition. PRICES ARE NEGOTIABLE!!

CONTACT: ANDREW BONG, BOL 227, QUIGLEY, CA 95444 (707) 823-1227

FOR SALE

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 OLD DEFUNCT COMPUTER STORE. 1-P-TECH 2KIO KIT, 1-P-TECH 2KIO ASSEMBLED, 3-P-TECH 16K SEMI KITS, 1-VOM-1 ASSEMBLED, 3-ALS CASSETTES & MANUALS, 2-MUSIC SYSTEM CASSETTES AND MANUALS, 4-EXTENDED BASIC CASSETTES, 1-5K BASIC CASSETTE, 1-MSD VIDEO BOARD 2480 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 CROMERCO. CALL DENNIS (916) 443-4944. 2325 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 Density Disk Controller and two Shugart 400 Drives are 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).

361 Watsonia Drive
Foster City, California 94404
415-349-3622.
TABLE OF CONTENTS

SOFTWARE TUTORIAL - THE UCSD P-SYSTEM: A UNIVERSAL OPERATING SYSTEM by Stan Sokolow........................................... 1
WHAT'S NEW? by Stan Sokolow........................................... 1
MODIFIED HYTYPE I PRINTER DRIVER by Dean Mazur.................. 6
DOCUMENTATION FOR "FORM" FILE by Allen Fincher.................. 8
BULLETIN SYSTEM FOR PROTEUS VIA RRBS by Alastair Preston....... 9
SUPER-PHANTOM FOR SOL-20 by Fr. Thomas McGeehee................. 9
LETTERS TO THE EDITOR:
...ON MCFLOPPY, MCSOLORS, AND ELECTRIC PENCIL by R. Downs..... 12
...CON'T FROM PAGE 8: DOC FOR "FORM" FILE by J. Fincher........ 13
PASCALSET TOOLS FROM I 1ST SYSTEMS CORP. by Stan Sokolow....... 13
...ON THE EXPANDORAM, ECBASIC TO CP/M CONVERSION, MICRO
...COMPLEX VIDEO BOARD by John Barber.............................. 14
...METHOD WANTED FOR CATALOGING HELIOS DISKS by F. Sanders... 14
...ON BOOTING UP THE SOL WITH NORTHSTAR by D.W. Stowell...... 15
...JOIN THE NEVADA COBOL USERS GROUP by Bob Blum................ 15
..."SWORD": A MINI-WORDPROCESSOR FOR THE SOL by Dennis Fait... 15
...ADDITIONS FEATURES TO AN UPGRADED SOL by Chuck Hansen..... 16
...A SUPER PRINTER FOR THE BUCKS by Paul W. Kittle............... 16
...HELP WANTED FOR REPAIRING SOLS by NonComputing Systems..... 16
...A RECOMMENDATION FOR STETSON ELECTRONICS by John Whiting... 17
...REPLY TO LEONARD KALISH VISTA CILOSE KEYBOARD INPUT
...PROBLEM by Allen T. Fincher...................................... 17
...CON'T FROM PAGE 1: WHAT'S NEW by Stan Sokolow.................. 18
...ON CONNECTING THE EPROM MX-80 TO THE SOL by R. Shuklin...... 18
UNCLASSIFIED ADS.................................................... 19

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 Pear 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) 388-2300.

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.

Copyright (C) 1992 by Proteus. All rights reserved. Permission is hereby granted to reproduce any computer programs contained herein, provided that Proteus and the program's author are given credit.

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

FIRST CLASS MAIL

Stephen Maguire
2430 N. Dodge Blvd
University of Arizona
Tucson
85711-1032
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, let's 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.)

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 connectors 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 286 microprocessor and ROM firmware to take care of sending and receiving packets of data, verifying and acknowledging their accuracy using error-detection data. Each packet utilizes the host node or destination node is able to receive them, etc.

There are many different types of computers, including those using the S-100 bus, the Intel Multibus, the DEC Q-bus (LST-11), the UniBus, the Apple bus, the IBM Personal Computer bus, etc. Additionally, there are a wide variety of computers that 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 a number of 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 an own disk. At the lower left, first is a "communication system" 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, such as packet-switched versus circuit-switched and connect with another local network that is some distance away.

The diagram is more complicated than any real application would be, but it illustrates the possible complexity and layers of a network. For example, a user on the computer in the upper right could use files located on the disk connected to his own computer. Files could be accessed remotely and files to a remote computer. 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 network is essential. But in a single office, it may still make sense to use a local network. Networking several small computers may be a better system for a small office than using a more powerful timeshared computer and with several terminals. Can you think of a reason? Well, here are a few.

The timeshared system is more vulnerable to catastrophic failures. If the central computer becomes inoperative, the entiremodel 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 backlogging this would create in a business that is all done "online"?

Worse would be to have local computing power at each workstation, so that data can still be gathered for later transmission to a malfunctioning node. Some programs require having all the data in one place, and 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

In a timesharing computer, the more users that the system supports, the more the system response time degrades. Even if a fast computer can only be shared 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 enough for it to be usable for one of the nodes needed by the user. But even then, the response for locally processed tasks is slow. 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 node. This way, the node can still perform tasks. In a timesharing computer, increasing the computing power generally means replacing the computer with a faster one, and this generally means a larger increase in cost. In a network, more nodes can be added 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 a new system to be scrapped. Therefore, 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 inventors of CP/M operating system) has developed its own networking software called CP/MNET and IP/MNET. These allow node computers to communicate with each other, if necessary, by using an interface hardware, to send messages, use data links, share printers, etc. Destek is developing its own network operating system which is designed for a wide range of applications. The International Standards Organization (ISO) has developed a model of network software to facilitate more standardization. They depict the possible layers and functions of a network. At the most primitive level is the Physical Link Layer. This is the hardware of the MAU. The IP/MNET layers are software 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 into smaller tasks, 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 a few more things to learn about: collision detection, error correction, protocols, device independence, queuing, 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-2 1.2 EQ North's Mixer 1/4" disk system? I have followed all of SORMUT instructions - IT BOMBS UP!!!

Please call: Millard F. McKinney
33137 Isle Royale
Fremont, CA 94538
(415) 656-7939
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 through 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-originated 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.

Jeffrey Sinclair

\*PRTRIP - in serial port of SOL and out parallel
\*Buffer routine
\*Contributed by Wordcraft
cout equ 0C054h ;VDMOT in SOLOS
esc equ 18h
space equ 20h
or equ 0Dh
lf equ OAh
feed 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 init
    mov b,a
    call cout
    inx h
    jmp msgout

init1 nup
  "Main loop:
loop call serchk ;check input, get if there
    call prtcchk ;give printer a character if ready
    call kybdchk ;let user pause or command
    jmp loop

serchk in 0F8h ;serial status port
    and 40h ;received data mask
    rz
    in 0F9h ;get data
    lld impr
mov m.a, ;buffer the character
    call incr ;increment input pointer
    shl impr
    ret

prtcchk call comprs ;compare in and out pointers
    rz ;have printed everything in buffer
    in 0F9h ;parallel status
    ami a ;printer ready mask, active low
    rmz ;not ready
    lld outptr
    mov a,m ;get char to print
    out 0F9h ;parallel port
    call incr ; bump the out pointer
    shr outptr
    cpi feed
    rmz
    lda pf1 ;user can toggle this flag
    ora a
    rz ;pause flag is down
    call halt ;wait for a space
    ret

kybdchk call keystat
    rmz ;no key
    call keyin
    cpi space
    ez halt
    ret

halt call serchk ;check input while paused
    call keystat
    jmz halt
    call keyin
    cpi space ;toggle space bar
    rz
    jmp halt
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 PDOS 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 debugging, 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 systems 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 reductions of their popular CP/M-based software series. Nevada COBOL, Nevada FORTRAN, Nevada PILOT, and Nevada EDIT have all been reduced to $259.95 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 Softtech Microsystems>

IBM SELECTS SOFTECH'S USC-D-P

IBM has selected SoftTech Microsystem's USC-D-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 USC-D-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 USC-D-P has not also been used, most notably in Software 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 his 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 San:

Enclosed with this letter is documentation of ABLE, a line editor written for the Sol-20. ABLE requires at minimum a 90-column screen & 360K Sol-20 disk with at least one CP/M-based disk. It's at its best with the 132-column screen upgrade and 540K 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 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 ASSIGN/KILL 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.

Years 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 Edit. 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, or CRT 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 possible 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*. 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 R, give the user immediate access to the top or bottom of the text file. No hidden toggles are used (example: the page down (SCRD) key is different than the page up (SCRU) 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 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 handle 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

NAME = 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 FS 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.
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 saving. 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.

PM (File Merge) — Merges a file to any place in the work area. After the command, FN, you'll be asked for the merge file name. Then you will be asked in front of which line number the file is to be merged. EXAMPLE: If you indicate line #5, 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.

CRLS — Displays the NEXT relative line. On loading a file, the current line is line #0. Typing CRLS will display line #1.

L[line number]CRLS — Alone, L shows the current line. If followed immediately by a number, L will display the line at that line number.

D[line number]CRLS — 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.

$<$CRLS — 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.

"CRLS — Displays prior line group, relatively to the current line. Group size is set with the V command.

Y[number of lines]CRLS — Sets the number of lines displayed in a group display command. E.g., D,8, or " commands will display a 5-line group after the command VS. This command is very useful to display two or more narrow groups of text adjacent to each other on the screen.

T(CRLS) — Moves you to the Top of the text area (line 0). Note: Line 0 never contains text. So D,1 do not reveal anything until you've pressed CR or 8<CRLS> to advance the display into the text area.
**BCRD** — B is for Bottom. Moves you to the last line in the text.

**T[ln]C** — 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.

**ACCRD** — Append a line to the end of the file. If the last line of the file is 99, then ACCRD will create line 100 and prompt the user with the edit prompt. You need only to type in new characters and type CCRD 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.

**[line number]C** — If the optional line number follows the command, 1, then you are inserting the line AHEAD of the line number shown in the command. If you enter 1 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 form feeds 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 form feed, type CCRD at the line desired. When you get the empty line and edit prompt, type CTRL-L (formfeed) followed by a carriage return to end the line. A clean screen character (CTRL-X on the Sol-20) may be inserted in the same fashion. You may use EDIT's PIP with the F extension to remove form feeds from the text file by copying it into a new file. Example: PIP FILETM = FILEONE[F].

**[line number]C** — If used alone, "M" 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, "M" 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 hang in another "M". ABLE always buffers the very last line killed, by the way. If you want the line restored, type "P" and the target line number.

**C** — Creates a new text area ready for use. Blanks all existing text. Be sure you have any needed text before using the C command.

**M[ln]** — Move a line. [ln] 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[ln]."

**C[ln]** — Copy a line. The same as M[ln] except that the line is copied, not removed from its original place. Type C[ln] to buffer the line, type P[ln] 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[ln] and P (no number).

**P[ln]** — Place a line buffered by M or C commands. "P" alone means put it after the current line displayed. P[ln] means put it at the line indicated by [ln].

**ENTERING AND USING EDIT MODE**

Edit mode has its own commands. The initial stands for what they do, like s for search, x for extend, h for hack, and so on. CCRD gets you out of edit and back to command mode. Enter edit mode by typing:

**[line number]C** — "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\slash diget".

**X(character)** — Deletes all characters from the cursor up to but not including the character typed after "X". The deleted area is surrounded by reverse slashes, as in "The\slash quick\slash brown\slash 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, K, 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, K, 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 CCRD). You may delete characters added merely by touching the DEL key; the cursor will move backwards and the characters will be rubbished 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 CCR 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 ARLS

ARLS -- 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";

The ARLS user may break a long BASIC=80 logical line into several physical lines by 1) 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 2) by embedding an underline character at the end of the physical line and closing the line with a carriage return. Method 1) causes the logical line to stay within one ARLS physical line, which has a maximum length of 255 characters, offering more room than the 128-byte limit of the BASIC=80 compiler.

Method 2), the underline (CR) termination, may be used if logical line lengths are longer than 255 characters. Method 1) will be readily visible to the programmer, while Method 2) 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 unusual to create and repeatedly 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 this, it would be wise 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)549-6873
30 40 PRINT"";"";"";"";"";"";"";"";"";"";"";"";""
40 PRINT"";"";"";"";"";"";"";"";"";"";"";""
50 INPUT"";"";"";"";"";"";"";"";"";"";"";""
60 INPUT"";"";"";"";"";"";"";"";"";"";"";""
70 OPEN "";"";"";""
80 OPEN "";"";"";"";"";"";"";"";"";"";"";""
90 
100 WHILE 1
110 1 LINE INPUT #1,A$ 
120 IF EOF(1) THEN 180 
130 GOSUB 210:"";"";"";""

140 PRINT #2,A$ 
150 180 
170 REM STRIP LAST LINE AND ADD " " MARK FOR CP/M EOF MARKER 
180 GOSUB 210:PRINT #2,A$:PRINT #2,CHR$(26):CLOSE 
190 210 
200 
210 REM SUBROUTINE STRIP LEADING NUMERICS 
220 FOR IDX = 1 TO LEN(A$) 
230 CHS = MID$(A$,IDX,1) 
240 IF CHS >= "0" OR CHS <= "9" THEN 260 
250 NEXT 
260 A$ = RIGHT$(A$,LEN(A$)-(IDX)) 
270 RETURN

THE SAME PROGRAM "STRIPPED, PIPPED AND ABLE"

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 lowercase. The command to do this would be: PIP LEASURING.BAS|STRIPPED.BAS|...

REM BASIC=80 LINE NUMBER STRIPPER 
REM (c) 1982 Daniel S. Hunt 714)549-6873 
REM "BASIC=80 source line number stripper" : print 
REM input"source file";fA$ 
REM "destination file";fB$ 
OPEN "";"",fA$ 
OPEN "";"",fB$ 

' while -1 
 1 line input #1,A$ 
 2 IF EOF(1) THEN 180 
 3 gosub 210:"";"";"";""
 4 print #2,A$ 
 5 wend 
 6 180 rem strip last line and add " " mark for cp/m eof marker 
 7 gosub 210:print #2,A$:print #2,chr$(26):close 
 8 end 
 9 210 rem subroutine to strip leading numerics 
 10 for idx = 1 to len(a$) 
 11 chs = mid$(a$,idx,1) 
 12 if chs >= "0" or chs <= "9" then 260 
 13 next 
 14 a$ = right$(a$,len(a$)-(idx)) 
 15 return

Note that the IF-OR-THEN construct just above line 260 in the stripped, lower-case version of the program was edited with ARLS'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. ARLS 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.
ABUS's EDITING ALGORITHM --- Editing a line involves splitting an edit string into two stacks. This makes it 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 null. It is empty. As the cursor moves right in the file, it looks at the topmost (leftmost) character in FREE, does whatever is needed in the way of adding, inserting, changing or deleting it, and puts it off the FREE stack with a LEFTS function. If the character survives scrutiny, it is placed on the top (right) of FREE stack, which will continue to grow as the cursor moves 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 simply a manner.

Due to BASIC's method of garbage collection, ABUS exhibits a few strange quirks. One of these is an occasional pause as ABUS 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 ABUS, but can last a few seconds in the interpreted version.

ABUS's STATISTICS
ABUS.COM file size: 246
Maximum line length: 255
Language: Native code BASIC-80
Suggested minimum system: 68K CP/M (Edits 18K file)
Next system: 88K CP/M (Edits 28K file)
Configuration of this demo:
Clear screen character: CHR$(11) (So1=20/VIN)
Delete character: CHR$(127) (So1=20/VIN)
Stop character: CHR$(1) + CHR$(32) + CHR$(1) (So1=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.

Dental S. Hunt
Green Valley Software
359 Princeton Drive
Costa Mesa, CA 92625
(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.

SUN RESEARCH INC.
Box 210 Old Bay Road
New Durham, N.H. 03855
(603) 859-7110 TWX 510-297-4444

...protection you can afford!
**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

Surcharging time: 40ms

"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.

**MAYDAY60+2C** .................................. 250 watts ........... $ 800.00
**MAYDAY60+6C** .................................. 600 watts ........... 1320.00
**MAYDAY60+10C** .................................. 1000 watts ......... 2295.00

BATTERY PACK

<table>
<thead>
<tr>
<th>MODEL</th>
<th>(VA) WATTS</th>
<th>LINE SURGE PROTECTION</th>
<th>DIMENSIONS (INCHES)</th>
<th>WEIGHT</th>
<th>BATTERIES (included with system)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M50</td>
<td>150</td>
<td>NO</td>
<td>11 x 12 x 7</td>
<td>23 lbs</td>
<td>1</td>
</tr>
<tr>
<td>M50-5</td>
<td>150</td>
<td>YES</td>
<td>14 x 10 x 8</td>
<td>34 lbs</td>
<td>1</td>
</tr>
<tr>
<td>M50-2</td>
<td>250</td>
<td>NO</td>
<td>14 x 10 x 8</td>
<td>38 lbs</td>
<td>1</td>
</tr>
<tr>
<td>M50-2S</td>
<td>250</td>
<td>YES</td>
<td>14 x 10 x 8</td>
<td>52 lbs</td>
<td>1</td>
</tr>
<tr>
<td>M50-6</td>
<td>600</td>
<td>NO</td>
<td>14 x 10 x 8</td>
<td>42 lbs</td>
<td>2</td>
</tr>
<tr>
<td>M50-6S</td>
<td>600</td>
<td>YES</td>
<td>14 x 10 x 8</td>
<td>63 lbs</td>
<td>2</td>
</tr>
<tr>
<td>M50-10</td>
<td>1000</td>
<td>NO</td>
<td>17 x 21 x 9</td>
<td>90 lbs</td>
<td>4</td>
</tr>
</tbody>
</table>

MAYDAY for 220 volt 50/60Hz operations are available upon request.

Prices effective March 10, 1982

Total harmonic distortion typically 3% on 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. 03895

(603) 859-7110  TWX 510-297-4444
The service test disk contains 10 programs intended to aid the service technician test Sol, 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 carriage return. The following is a brief summary of the available tests:

16KRA: not used

Tests a 16K block of contiguous memory beginning at 0000 to 00FF 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 to 00FF 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 to 00FF 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 00FF 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 - 00FF in 4K increments writing to each block a total of 9 times. The test will either stop on errors or continue on errors but in either case will report the failure. Pressing the "Mode Select" key will return the user to PTDOs.

SIMU:

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.

DISK:

This program is the DISK diagnostic program detailed in the Service and Maintenance manual and the HELIOS manual. The test is located from 0000 to 00FF and uses 3000 - 3FFF as its buffer. The program is executed at location 0003.

DISKCHK:

This test is 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 run. After 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 error reported will be PTDOs errors since this test runs under PTDOs control.

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 run. After 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 error reported will be PTDOs errors since this test runs under PTDOs control.
I wrote to you several months ago about the Electric Pencil (possibly PROTRUS 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 right to Pencil, and it appears that they either don’t 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 CPM, so that the cassette I/O commands do disk I/O instead. The necessary modifications consist of relabeling 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 PROTRUS?

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 at 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 Duemer

F. O. Box 1451
Homewood, IL 60430
September 12, 1982

(Editor’s reply to John Osunder)

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 280 processor. It appears that more and more software is being developed for the 280.

I would appreciate any information regarding the above.

Sincerely,

Michael C. Mogdan
Michael C. Mogdan
26 Walden Lane South
Burnsville, MINN 55337
October 5, 1982

EDITOR’S NOTE:
We have a Z-80 upgrade module in final 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 5/5:

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 Hogg Solos. That is the correct initialization address at address 02A4. If this is 00009 then no change is needed, but if there is a call to SOL’s clear screen in the *CM 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, 605 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 built in initialization program that works for the SOL or a CUTS board with YMM.

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 CBASIC 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 also basic programming that you make as interesting display to accompany your educational programs. There are even commands to let you control video 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 personnel in using that new computer you just introduced. 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

---

**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 applications 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 Rowler, 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

Rowler expects the 8-bit market to remain strong through the 1990s, citing as evidence the appearance of National Semiconductor CMOS 280 equivalent devices, the Zilog Z800 and the Intel 10 rewriter. "CP/M Plus offers at least one disk and a console. The non-banked version will require a minimum of two banks with the top 16K (or 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 in product literature from Digital Research Customer Service.

An overview of CPA 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.*
Calculated under identifying conditions for disk file

Don't type this EVER again

Now...the first super program that puts YOU in control of CP/M.

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 the 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 mistakes. Furthermore, the program traps that unwelcome DDOS error...more importantly, it allows you to correct errors with a reclain function that restores accidentally erased files.

Power makes CP/M a joy to use.
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 ALGB0, 0 to 4KB, for SOL without ALGB0 0 to 4KB, and 52KB to 64KB). Has invisible refresh and does DMA (board has been tested with UUGBOUT DISK 1 DMA Disk Controller). Latest rev. P.O.B. Complete with manual for only $500, plus tax (Cal. residents 6.5%) and shipping. Limited supply. Contact Bob Demaret, 396 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.


All 3 above for $50 plus tax and shipping.

Very limited supply of delay lines.

Contact Bob Demaret, 386 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, Expanded Cassette BASIC, no monitor.

Charles H. Beineman
309 Cambrone 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 R6 212 with tractor. Has SOL Rev. D with 48K memory, PTBOS 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 (PTBOS, languages, games, utilities, etc). Will sell the board sets and drives separately. $250.00 for the board sets, $375.00 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 48K 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 WOII/WINT
G.P.O. Box 71390
San Juan, Puerto Rico 00936

FOR SALE - Helios 2 Slot w/PTBOS, including BASIC software, Mario Pammatan, 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 Cambrone Dr., Englewood, OH 45322 (513) 816-6054.

WANTED: SOL P.C. board, preferably in working condition, but I will consider any condition. Jim Williams, 2246 Cowan Blvd. Apt 304, Fredericksburg, VA 22401 daye (703) 371-8316, evenin (703) 373-2569.

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. $400.00 or make me an offer. Royce D. Bacon, 8942 W. Lawrence Ave., Milwaukee, WI 53225, phone (414) 662-3418.

EMERGENCY - Looking for Per Sol 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 Viejo, CA 92691, (714) 770-1168.
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 it). But by now, there is such a bombardment of information about microcomputers and such an overwhelming onslaught 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, Impal, 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 2-80 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 "8086: 16-bit chip on the 8-bit bus" craze, especially with IBM entering the personal computer market with an 8086 computer. But, benchmarks have shown the 8086 is not so swift with the software out there now. My Sol/Mellon 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.
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.

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 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 human-friendly menu where users are totally unaware of the underlying p-System. (In other words, absolutely no jargon will appear on the screen to confuse the user.)

A performance monitor allows the user to track such automatic system activities as retrieval and printing of code segments from

US POSTAL SERVICE OFFERS ELECTRONIC ORIGINATED MAIL

E-COM (Electronic Computer Originated Mail) is a new service offered by 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, encolved, and delivered 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 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 letter. This includes printing, paper, envelope, and postage. There is also a minimum 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 (CTM), and text insertion messages (TIM). Each SAM's include the text and destination address for each letter. CTM's contain one block of text to be sent to each of many addresses. TIM's allow plugging-in of information into a skeletal message created by the user. Each message includes an address and unique information are provided by the sender along with the skeletal form.

Text may be a 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, no single window envelopes are permitted in international mail, so I assume this is only available for domestic addresses.

E-COM is useful for 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
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 8085 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., you can 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.

How do you get all this new software? Not from 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 your 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. There's a big step forward in ease of 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 now.

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 new model or space? We are in computers, it may happen that every machine will provide an 8080/8085/5280 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 runs on various microprocessors and 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 instruction in each computer that uses the p-System contains an interpreter.

UCSD p-SYSTEM FOR HELIOS

CON'T FROM PAGE 1 (US POSTAL SERVICE OFFERS ELECTRONIC ORIGIRNATED 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.

program that reads the p-code instructions and simulates the p-code instructions and simulates the p-code 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 interpreted. 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 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, and not the University 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 2-80'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 canonical 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: 8080, 8086, 8088, 280/8080, PDP 11 (various models), LSI-11, 6502, and 9900. It was aimed at covering all of the current machines, even the new Apple Lisa which uses a 68000.

So, although I love PTDOO on my Helios, and although CP/M is ubiquitous and cheap and I have it already, I have installed the 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 Princeton Tech before he went on to the software integration 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 newsletter. 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 my earlier versions of the p-System, the third version IV.1 is an excellent system and definitely here to stay. That's why I'm not enthused about working on transporting PTDOO to other microcomputers. Of course, 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.

I would like to hear from you for publication in the next newsletter.

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 my earlier versions of the p-System, the third version IV.1 is an excellent system and definitely here to stay.

That's why I'm not enthused about working on transporting PTDOO to other microcomputers. Of course, 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.

I would like to hear from you for publication in the next newsletter.
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, the disk is used only for storage. The program could be adapted to run under PTC Basic or other bases without much difficulty.

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 received 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 doesn't making 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 disks.

Thank you,

Very truly yours,

Franz Hirner
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 DES 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 an existing spreadsheet. All numerical data and parameter data are automatically shifted and recalculated. This saves much time when revising an existing spreadsheet. Although the Parameter definitions are changed and data element references are revised, the user should use care to check the Parameters when 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 scroll rate.

Command: (A) (D)
Analyzes the spreadsheet and calculates those numeric values which have been defined by Parameter as a calculated value derived from other elements in the data array. The program examines every data element on the spreadsheet proceeds 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 elements, that those data elements will in fact contain the expected values at the time of calculation. Single spreadsheets with few data elements and few calculations will take just a few seconds to analyze. Larger spreadsheets with many calculations will take more time, depending on the size of the 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: (E) (D, P)
Allows the user to select the area of the spreadsheet 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: (F) (D, P)
Allows the user to enter either numerical or parameter data depending on the current display screen. The user will be prompted for the row, column, and input value. If in the...
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. If the first Parameter is entered, 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 program.

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) examines 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 (T) 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 data 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 than 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 (similar to C for Columns).

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 specific data elements.

AK
Add Constant Adds the constant contained in Variable 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 FROM PAGE 5

numbers whose significance to the printed output is minimal should be stored in this fashion.

CO

The cosine of the 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 Variable 1. The result 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 R1-C2. The user must make sure that either R1 and R2 are identical or that C1 and C2 are identical. 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.

GI

Get Inverse is similar 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 Variable and should be entered as a whole number, i.e., 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, then the total amount of payments for the year is entered, and the yearly interest rate expressed as a percent is entered. The variable Z2 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 an 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 Variables to the data element referenced by R1-C1. The result is stored in the home data element.

PG

Percent Change is similar 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 100 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 i.e., 50%. The result is stored in the home data element.

SK

Subtract Constant subtracts the constant contained in Variables 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
| Page 9 |
INSTRUCTIONS FOR RELOCATING ELECTRIC PENCIL I VERSION SS
TO ANY PAGE BOUNDARY

by John Duda
P.O. Box 1451
Homerill, IL 61440-0451

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
I have relocated MY copy of EP successfully to origins of 0000H, 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 available—one
in Leonard Zelman'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
Leonard Zelman'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 either of these, he has a dump of Leonard Zelman's
program supplied with these instructions.

WHAT YOU DO

(1) Load the ELECTRIC PENCIL program from cassette or disk into memory
starting at a new destination. (e.g. relocating to a new origin of
0000H, so a GET PENCIL 300 from ZLOOS, or the equivalent in your DOS.) Note
that PENCL will occupy a total of 0500 bytes of memory -- keep this in mind,
so that you don't overwrite your DOS, or the relocatable program, with the
loaded PENCL (or vice versa).

(2) Load the relocatable program into a safe area of memory. (Safe = not
occupied by operating system, PENCL, or other active code or data.)

(3) The relocatable programs require six parameters, of which four remain
unchanged throughout the relocation. For Leonard Zelman'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. A function code
   0 for fixing references only.
   1 for moving and fixing references.
   2 for moving only.

The relocatable 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
function-fixing function.
The parameters to be entered to relocate PENCL are as follows: Call 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 Leonard Zelman's program:
A = 0001 01 00
B = 00FF 0F 0F
C = 0001 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 "0000"
ADDED TO IT (OR "xx" ADDED TO THE SECOND (HIGH-ORDER) BYTE;
For example, if you are moving PENCL to an origin of 0300H, 0100H,
0200H and 0300H. If you have any additional information, please send it to me
at the above address.

<table>
<thead>
<tr>
<th>D value</th>
<th>E values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0000</td>
<td>004F</td>
</tr>
<tr>
<td>0053</td>
<td>01BF</td>
</tr>
<tr>
<td>01C2</td>
<td>022F</td>
</tr>
<tr>
<td>0333</td>
<td>0257</td>
</tr>
<tr>
<td>026B</td>
<td>0508</td>
</tr>
<tr>
<td>050C</td>
<td>0542</td>
</tr>
<tr>
<td>05A4</td>
<td>05D3</td>
</tr>
<tr>
<td>05D7</td>
<td>0705</td>
</tr>
<tr>
<td>0709</td>
<td>0750</td>
</tr>
<tr>
<td>0709</td>
<td>076C</td>
</tr>
<tr>
<td>07CA</td>
<td>08BD</td>
</tr>
<tr>
<td>08C1</td>
<td>0A13</td>
</tr>
<tr>
<td>0A17</td>
<td>0C53</td>
</tr>
</tbody>
</table>

(4) You must now make one change manually. At address 01C1=xx00 you should
find a byte containing 06H. Add the value xxH (i.e. the high byte of the base
address) to it.

(5) Your Electric Pencil Version SS has been relocated to a new origin of
0000H. Save it on cassette or disk and try executing it from PENCL (with an
EX xx00 command). It should operate just as it did when located at 0200H. 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 use the stack and registers similar to the way SOLOS does
before jumping to PENCL. This is why I would make a CF/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 BP pointing to a usable stack area with the
SOLOS return address on the stack already. You can emulate this by doing the
following instruction sequence
LXI BP, C000H
LXI H, 0200H
CALL xx00H
JMP C004H
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 nonresident memory or the first byte
of read-only memory.) If you have your operating system set the the
PENCL somewhere (e.g. CF/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
000E through 002E, a total of ten bytes, and the last available address is stored
at 000A=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
clarament zeroed.)

CON'T ON PAGE 11
A word of warning: PENCIL starts clearing memory at 0DF7H+8000H, as should your patch routine. This, however, causes you to zero out the contents of PENCIL’s stack. PENCIL doesn’t have anything on the stack when it is doing the initial direct return, neither should you. 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 0E00H+8000H after you clear memory, for the same reason.

You will also need to patch a loop at 0A05H+8000H through 0A09H+8000H, which is used by PENCIL’s CAA and CAS commands. Entry at 0A05H+8000H is with the first address to be cleared contained in HL. PENCIL does not appear to care what’s in the registers when you were done, as long as the SF is still the same when you hit the RET instruction at 0A05H+8000H.

Please note that at this point you have only moved PENCIL. PENCIL will still NOT work with your disk operating system (i.e. 8000H).. It will still expect a cassette recorder to be used as the file device.) Getting it to do that is far more complicated, but it can be done — my PENCIL uses my North Star DOS for disk I/O. The way I chose to implement this is based upon the fact that PENCIL 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 0000H-FFFFH for a “fake SOLOS” in which the cassette I/O entity is replaced by new DOS disk I/O) and load the start address of this “fake SOLOS” into HL before calling PENCIL, then PENCIL will use the entry points in your version of SOLOS. It uses the standard entry point temp table, so it’s easy to put in disk I/O routines, you only have to emulate the cassette block I/O entry points RXBLK and RXBLK, 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 RXBLK and RXBLK routines!

LEOZDMAN’S MACHINE CODE RELOCATOR FOR THE 8080
(This program was originally printed in the October 1777 issue of BYTE magazine. It has been modified to jump to SOLOS at 0000H instead of infinitely looping when done.)

Enter this program on a page boundary (i.e. 8000H)

The following is a hex dump of the program, sixteen bytes to a line. Note that “sz” represents the high byte of the starting address throughout.

31 FF 2A DF 54 5D 2A E1 44 4D 2A DD 3C
1A 77 78 B7 C2 EC 79 B7 C3 31 2B 1B 0B C3
20 82 2A E7 F8 02 CA 0C 86 62 68 CD 01
D1 19 22 8E 02 2A E3 2B 1E 3B 0B 78 78 79
73 7A 9C DA 04 0C 06 1A 11 E1 1A EA CA 7B 08
08 2C 5B 06 12 11 CB 1E 1A BE 9A 01
3C 2A 4A 5A 2C 09 23 C3 49 3E 02 2A DF 54
5D 2A DD DD 44 4D E1 23 7B 94 2C 7A DA 49 29
3E 76 71 13 7E PA 49 3E 2E 0A 8E 8A 06 0E 7E
B3 77 23 7F 8A 00 79 C4 49 57 2F 47 7F 02 6F 23
C9 01 11 21 22 3A 31 52 3A C3 C3 C4 0A C3 CD D2
D4 DA 8E 02 02 EA EC F2 04 0B FC F9 0E 14 1E 24
2E 36 3E C6 CK DD 06 DE E6 EE FE

The next 11 bytes are used for parameter storage, as follows:

ss5D, ss5E: First address of block to be relocated
ss56, ss57: Last address of block to be relocated
ss58, ss59: Destination address routine of first byte
ss5A, ss5B: First address to have references fixed after move
ss5C, ss5D: Last address to have references fixed
ss5E: Function code
00 = fix references only
01 = move block and fix references
02 = move block only

The two bytes at ss5E and ss5F are used internally for storage.

Ordinarily, only the values of ss5E through ss5F are used. These may be modified when relocating parts of a large program (like ALIB or PENCIL).

NOTES ON A CP/M INTERFACE FOR ELECTRIC PENCIL VERSION 85:

The attached 8080 assembly language program, PENCIL0, is a CP/M interface for Electric Pencil Version 85. 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.

PENCIL0 works by feeding PENCIL into thinking that SOLOS is located at 0100H. This is done through the value passed to PENCIL in HL when PENCIL 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 arc the SOLOS-in-ROM addresses (e.g. 0107H contains a JMP 007CH). The only entry point that is changed is INIT, which does some setup. Now, if the call to RETN, which jumps to 0 to re-boot CP/M, and RXBLK/RXBLK, which do disk I/O instead of cassette tape I/O.

The functionality provided by PENCIL0 is almost identical to that provided in the cassette version. Files are named with five characters internally (e.g., ABDDE), these names are translated to CP/M filenames by first replacing spaces and/or control characters with a Dash (-), then extending the name to its full length by appending -.EP.TXT, so that ABDDE is stored on disk as ABDDE-EP.TXT. PENCIL0 will rename an existing file of the same name to .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 PENCIL0 itself. Units 1 and 2 are implemented to refer to A and B, and it will be changed by modifying PENCIL0’s MAKEFILE routine to select other units.

SUMMARY OF STEPS IN IMPLEMENTING ELECTRIC PENCIL I VERSION 85 UNDER CP/M

(1) Relocate PENCIL to 0300H
(2) SAVE 16 PENCIL.COM in CP/M
(3) Enter and assemble PENCIL.COM
(4) Test PENCIL.COM
(5) In DET, do IF PENCIL0.HEX

R to read PENCIL0 into memory
(4) Exit DET with AC or GS
(7) SAVE 16 PENCIL.COM again
(8) Now, PENCIL should get you into CP/M PENCIL!

NOTE: If you have moved your SOLOS and video memory to addresses other than 0C000H and 0C8000, respectively, then you will have to modify PENCIL.COM to point to the new SOLOS, and you will have to make the following changes to the relocated PENCIL.

At address 0DF7H+8000H (i.e., at 10F4H if you have relocated PENCIL to an origin of 3000H) PENCIL 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. PENCIL will NOT work with the 24-by-80 modification at least this version won’t!

CONT’D ON PAGE 12
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 VOL. 06

WRITTEN 29-MAY-82 BY JOHN OSUDAR

THIS IS A PSEUDO-SOLOS THAT SUPPLIES THE RBLK AND WREBLK FUNCTIONS TO READ/WRITE CP/M FILES FROM THE ELECTRIC PENCIL VERSION SS. PENCIL MUST BE LOCATED AT ADDRESS 0000H OR ABOVE. AS THIS CODE OCCUPIES ADDRESSES 0100H TO 020FH.

ORC: 100H   START AT TPA START
SOLOS: EQU 0C000H   START OF SOLOS IN ROM
CPM: EQU 0300H   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, RBLK, AND WREBLK 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 :FCLOSE
JMP SOLOS+0DH :NOTABY
JMP SOLOS+10H :WBBYT
JMP FWRBLK :CP/M RBLK
JMP FWRBLK :WRBLK
JMP SOLOS+1FH :SOUT
JMP SOLOS+1CH :AOUT
JMP SOLOS+1FH :RINP
JMP SOLOS+21H :AINP

INITP :INITIALIZATION CODE

INITP: LIXI SP,SOLOS+0C00H :SET UP STACK IN SOLOS AREA
LIXI D,PATCH1COOD :PREPARE TO PATCH PENCIL
LIXI H,PENCIL+1EH :SET TO ADDRESS WHERE TO PATCH
MVI C,10 :TEN BYTES PATCHED

PATCH1: LOADX D :COPY LOOP
MOV M,A :
INX D :
INX H :
DCR C :
JNZ PATCH1 :
LIXI D,PATCH2COOD :SECOND PATCH
LIXI H,PENCIL+5A06H :SIX BYTES PATCHED

PATCH2: LOADX D :COPY LOOP
MOV M,A :
INX D :
INX H :
DCR C :
JNZ PATCH2 :
LIXI H,PSOLOS :POINT TO OUR JUMP TABLE
CALL PENCIL :CALL ELECTRIC PENCIL
JMP REBOOT :REBOOT UNTO RETURN

PTCH1COOD CALL MEMPATCH :EXACTLY TEN BYTES OF CODE
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP

PTCH2COOD CALL ZERPATCH :EXACTLY SIX BYTES OF CODE
NOP
NOP
NOP

MEMPATCH POP H :PATCH ROUTINE
SHLD RETURN
LHLD 6000H :SET BACK RETURN ADDRESS
LHLD CP/M START ADDRESS :STORE IT
LD D H :SUBTRACT 1EH
DAD B :STORE AS LAST AVAILABLE ADDR
LXI D,0DP7H+PENCIL :START OF AREA TO BE ZEROED
INCLR MOV A,H :NEGATE HL
CMA MOV M,A :
MOV M,A,L :
MOV M,A,H :
INX H :
XCHG :SWAP POINTER & COMPARISON
CLRLoop: MVI M,0 :CLEAR A BYTE
INX M :
JMP PUSH :ADVANCE POINTER
MVI H,0 :SAVE IT
DAD D :ADD .HIGHEST ADDRESS
POP H :RESTORE
JNC CLRLoop :LOGF UNTIL DONE
LHLD LAST :SET LAST ADDRESS
JMP HERE :RETURN FROM HERE
RETURN EQU HERE+1 :STORE RETURN LOCATION HERE
LAST: DS 2 :

ZERPATCH: ECHG :SET-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

MAKEFGB :MAKES .TXT FCB FROM SOLOS HEADER

MAKEFGB: LIXI D,5CH :POINT TO FCB AREA
ORA A :
TEST UNIT NUMBER
JP UNIT3 :
UNIT 1 IF HI BIT OFF
MVI A,1 :
JMP DOWUNIT :
DOWUNIT: STAX D :
SET UNIT IN FCB
INX D :
ADVANCE
LOOF1: MOV A,M :
COPY FILE NAME
MVI C,5 :
JMP CP/I +1 :
TRANSLATE SPACES/CURRENT
JNC NOP :
MVI A,1 :
SWAP POINTERS
NOSP: STAX D :
STORE CHAR
INX D :
ADVANCE
INX D :
COUNT DOWN
DCR C :
AND LOOP
JNZ LOOP1 :
XCHG :
SWAP POINTERS
JNZ M,8 :
MVI M,'p'
MVI M,'e'
MVI H,M :
MVI M,'e'
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

MAKEFCB:

MAKES BAK FCB FROM HEADER, THEN CALLS MAKEFCB1

MAKEFCB:

PUSH PSW
SAVE FOR MAKEFCB1
PUSH H
PUSH D, 4CH
LIKE ABOVE, BUT 2ND FCB
ORA A
JMP AUNIT1
JMP DOAUNIT
AUNIT2: MOV A, 2
DOAUNIT STAX D
D INX D
MOV C, 5
ALOOP: MOV A, 1
CPF ' 0 T1
JNC ANOSP
MVI A, 'A'
ANOSP STAX D
INX H
INX D
INX C
JNZ ALOOP1
XCHG 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
MOV M, A
POP H
RESTORE FOR MAKEFCB1
POP PSW
JMP MAKEFCB1
CALL MAKEFCB1 AND RETURN

PROBLK

CP/M READ BLOCK ROUTINE

PROMBLK:

PUSH D
SAVE ADDRESS OF BUFFER
CALL MAKEFCB1
MAKE FCB
MVI C, 0FH
OPEN FILE
LXI D, 5CH
CALL CFP
IF FAILED, RETURN
POP D
RESTORE POINTER
RCX IF FAILED TO OPEN, RETURN
XRA A
SET RECORD 0
STA 7CH
SET DMA ADDRESS
PUSH D
SAVE POINTER
CALL CFP
SET IT
MVI C, 14H
READ SEQUENTIAL
LXI H, 68H
DAD D
JMP
RDLOP
AND LOOP
RDONE: XCHG
NOTEND: MOV A, M
ORA A
DCX H
JS NOTEND
LOOK FOR FIRST ZERO BYTE
INX H
POINT TO BYTE AFTER IT
INX H
INX H
RET

PURBLK

CP/M WRITE BLOCK ROUTINE

PURBLK:

CALL MAKEFCB1
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
OF 80H
MOV A, L
ANI 80H
MOV L, A
PUSH H
SAVE SIZE AND POINTER
PUSH D
DELETE OLD .BAK FILE
LXI D, 4CH
MVI C, 13H
CALL CFP
LXI D, 5CH
CALL CFP
RENAME .TXT TO .BAK
MVI C, 13H
CALL CFP
LXI D, 5CH
CALL CFP
MAKE NEW .TXT FILE
MVI C, 14H
CALL CFP
SUCCESS!
POP H
RESTORE SIZE AND POINTER
POP H
IF FAILED, RETURN
SET RECORD 0
STA 7CH
ELSE RESAVE SIZE AND POINTER
WRLOOP:
PUSH H
S
MVI C, 1AH
SET DMA ADDRESS
CALL CFP
LXI D, 5CH
WRITE SEQUENTIAL
MVI C, 15H
CALL CFP
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
DAD B
ADVANCE POINTER
STAX 6CH
SWAP THEM BACK AGAIN
JMP RDLOP
WRLOOP
AND LOOP
DOCLOSE: MOV C, 15H
CLOSE FILE
CALL CFP
RET
DONE

END
A SUBROUTINE TO LOCATE SOLOS

Leonard Morenstein

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 (415-645-4518). They charged me $3, 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 disastrous, 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 FOOO. If FOOO can be incremented, then it is RAM, and SOLOS must be at C000. If FOOO 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 FOOO will contain 3. 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 000007DH. As written, the program will abort if SOLOS is not at F000. By deleting the last few commands, the North Star DOS becomes indifferent to the location of SOLOS.

0010 WHERE
0020 *
0030 A SUBROUTINE TO LOCATE SOLOS
0040 *
0050 * Leonard Morrenstein
0060 * 204 Rheem Blvd.
0070 * Moraga, CA 94556
0080 * December 1, 1982
0090 *
0100 * PERSONALIZATION
0110 DDS EDU 00000H Set to location of DDS
0120 *
0130 ****************************************************
0140 WHERE tests location of SOLOS. Returns Z if at
0150 F000, NZ if at C000
0160 ****************************************************
0170 WHERE HL=FOOOH Set HL=FO00
0180 INR M Try to increment FOO0
0190 MOV A,M Result to A-register
0200 DCR M Restore FOO0 to original
0210 CMP M Compare A with FOO0
0220 RNZ If unequal then SOLOS is
0230 not at FOO0
0240 BRA A Does FOO0 EQUAL A?
0250 RET If Z then SOLOS is at FOO0; if
0260 NZ then it's at C000
0270
On another note - let me say that for those SOL users who are about to buy SuperCalc II, that the NORTHSTAR Format disk version is written in 80x code. It will not run on the standard SOL. The solution to this dilemma is to purchase the standard 3" version - it is written in 80x 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" drive and North Star to download it for you. Also, SuperCalc works best with an 80x screen format and since I upgraded I had to have it modified for the Dual 80x4 video board. While at SOL's I gave them the name of an individual who will 'install' SuperCalc onto the 80x4 board. His name is Dan Hunt of Costa Mesa Co. and his phone number is (714) 546-8772. However, there is a price but Dan is very reasonable. He modified my drives 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 80x4 screen which allows more information to be displayed on some of the WordStar menus.

Last summer, after setting up 80x4 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 used it.

SuperUser Area for Lifesaver CP/M

(1) Screen Size - 04x16 or 06x14
(2) Page Length - number of printed lines/page
(3) New Page - automatic formfeed if desired
(4) SOLCOM - 06x0 or 06x0
(5) Video Toggling - List 16 lines and stores or displays lines as you set.
(6) Printer Toggling - Directs printer status ON/OFF
(7) Paper - Sheet or Punched: this can be changed under program control
(8) Mode Select Key - Return to SOLCOM
(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: Toggles auto-display feature
[b] Load/O: Loads data
[c] Load/F: Toggles printer on/off
[d] Load/S: Sets printer up for sheet paper
[e] Load/1: Sets printer up for fanfold paper
[f] Load/P: Sets printer up for fanfold paper and auto formatting
[g] Load/Mode Select: Returns to SQL
[h] Load/Load: Pauses screen operation

Mr. thanks to Mr. Tom McShane 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 Mr. Tom is planning a full review if not in this issue then in an upcoming one. For those SQL users who might be interested can contact Mr. Tom at San Diego Technical High School, 202 Union Ave., Paterson, NJ 07502.

Since writing last I have taken on a new job and spend a lot 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 SQL, 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 80. Are you planning on attending? If so I would certainly like to stop by and meet you and talk about the SQL.

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 Downs
Denver, CO
(303) 750-1938

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, Keesey Mori, 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 62 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 well 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.

Appends 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-code machine interpreter. It is also an excellent way to learn about advanced techniques in the Pascal language.

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 recent work I've been doing with the Sol.

The table also has information for burning new PROMs (097, 095, 098 & 040) for the 64/48 KRA-1. The addressing logic to make an 8 KRAM into a 64 KRA-1. Four of these PROMs are required (2 for a 8 KRA-1) to make the board a full 64K (4Kx). All four output lines should be probrammed low at the address locations given. All other locations are to remain high.

My original 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 original PROMs were 742437s which require pull up resistors on the 4 output lines. At the time I could not get these so I used 7422Rf which do not need external pull up. This board never even slightly noticed that 2 of the 4 address PROMs have different output configurations. So I left the external pull up on the card for all 4 address PROM locations.

The original dynamic RAMs were Mostek MX410s which are 8K fall outs from the 16K by my manufacturing process. These chips were really 16K, but the upper or lower half failed the makers quality control tests. The MX410s were no longer available so I bought sixteen Mostek MX410s (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 guess 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 that this program must be like. That is the listing I included with this letter. The program is the same as the required for the 8 KRAM but only 2 PROMs are used. You will notice that the table is not in a normal list form. I made it like the DIP Switch address select table. Thus, because the actual PROM address is one of that selected by switch (pull up on the input), I inverted the table. A moment looking will make it apparent.

HELP! Please. I'm already to go on making this a 64K card, but I have 2 missing pieces of data. The original manual I had for the 32 KRA-1 included four drawings for a plug-in header called a configuration module. This allowed PVG to use different dynamic RAMs and also to take care of addressing the RAMs. e.g. A6 could be used for Col 0 or something purpose, etc. With this module, the use of the upper or lower half of the RAM or both was controlled. I was in hopes that Volume 6 (memory boards) of the Encyclopedia Porcoresor Techniques 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 742275 PROM (056). This is called the STATE MACHINE. I made a listing of the code for the 32 KRA-1, but I don't know if it's the same in the 64K version. 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.

Sincerely,

[Signature]

W. Leon Winter
1794 8 Madison Ave
El Cajon, CA 92021 USA
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 development outside of financial and related fields. For games, scientific applications, casual small programs for aid in programming, and anything that one can do without too much hassle, Pearl is not the way to go. But for inventory control, data-base management of books, records, or the like, accounting (if one is an accountant, others should learn that too), the pros, mailing lists, etc., it can be a real time saver.

A lot of time and thought went into the development of Pearl III; there are Pearl I and II, but II combines the features of both and adds considerably to them; III is necessary if the applications desired are at all extensive. The required tools are CP/M, CBASIC, lots of memory, the 88K24 mod for Sol, and some basic programming skills; 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 basic code as needed to create the unique program desired, and eliminates most of the tedious code keying. One can say that Pearl is not something one masters at the first sitting; like most DBMS and development schemes one must almost learn a new language. It's, however, all quite logical, and once you become acquainted 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 cleaner Rev.2 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 develop 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 what 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 Basic I'm sure. I guess we just can't 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 FROM PAGE 18

a lot of merchandise. The database 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. Russel
La Habra, CA 90631
(213) 697-7820

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

Redwood Drive, Atherton CA. 94023.

Thank you,

Sincerely,

Dan Ruder

---

SUPERFILE/Paul Kittle

**** NAILITCHIN SIBERIANS ****
Paul and Bov Kittle
Box 1259
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 FYL, Inc. and want to say it works very nice on Sol. (I've got an old set of Microplex 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

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. I write 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 compatible

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 had had a minor

problem that would have been major if I weren't an old maintenance

man. The problem arose 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 $290.00)

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
computer services
to support
the architectural profession

January 29, 1983

Stan Sokolow
Proteus
1690 Woodside Road, Suite 219
Redwood City, CA 94063

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 are used. Characters with a 5x7 dot matrix with blank dots between for 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 bandwidth 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 Garcia 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. The 17.1 MHz 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 is often not stated. Also, different ads claim different bandwidths for what seems to be the same monitor. I've seen this happen with the NEC JB-1201 monitor. Another monitor I'm interested in is the USI Pi-3. It has a 24 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. McAfee's recent article on his Super Phantoms 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 reason for any reason to move it to CO0 for the 80 character display. There would be advantages to having a 64 character Solos at CO0 for PDOS and old software and an 80 character Solos, also at CO0 for both modified PDOS and CP/M or P-System. Solos could be switched out for non-PDOS 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 2161's? I know an earlier Proteus News had a contribution about modifying the 2708 module to use 2161's. Unfortunately, it wasn't clear to us who are ignorant but can follow clear instructions.

Could 2716 switching be accomplished using the two extra PCH port controlled flip-flops provided with the McVideo modification?

I enjoyed 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'S ON PAGE 21.
CON'T FROM PAGE 20

possible with a standard, unablated 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,

Michael A. McKevelly

[Ed. note: The McVideo 64/80 column video upgrade for Sol does not 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 kHz 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 Nogg 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, Softtech says 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 toward more accuracy. 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.]

.NEETING PTDO FILES TO OTHER COMPUTERS/M.Pulsifler

NATHANIEL PULSIFER & ASSOCIATES
Investment Management & Financial Planning

February 9, 1983

Mr. Stan Sokolow
Proteus News
1690 Woodside Road
Redwood City, CA 94061

Dear Stan,

How can PTDO files be transferred to other computer systems? Could we use a communications program, modems and the phone systems (providing the new set up is ST 100 bus compatible)? How about plugging something into my SOL?

NP/ms

[Editor's reply: Transferring files from PTDO 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 Sol 103 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 PTDO device driver routine to call the appropriate Solos entry point.

The sending Sol with PTDO running should be given the command *COPY source,driver
driver is the device name to be sent and driver is your device driver name, such as Sol1. 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 8105 would have to contain the driver for that port, of course. The command on the receiving computer would be something like:

AP: PIP filename:dev:
driver is the name you give to the received data and dev is the device name corresponding to your 8105's driver for that port.]
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 comes along with an inflationary rise. Since thought about writing a version of PENCIL-L in 8080 assembly language, the needs expressed in the letters provided the motivation to do something about it. After many years of translating and modifying the disassembled PENCIL-L version, I am now presenting this letter so that PENCIL-L is available from disk in blocks of three disk sectors as needed, plugging the blocks alternately in two buffers. Hence, the space occupied by the source statements is only 95K (a total of 253K bytes). The size of the program can now assemble virtually limited only by the size of the source code that can be written with the EDITOR in 52K RAM.

I will gladly donate the assembler language source for my version of Kravett's PENCIL to PROTEUS if there are enough North Star advocates among the membership. There is no problem in adapting the program to the single-drive disk systems with DOS in its "standard" memory range 0000H - 1FFFH.

As indicated above, I now own a customized version of the object code and the assembler language source should let me know. A price of $20 would cover the cost for my time, a floppy disk, and postage.

Aside from the above, I would appreciate knowing whether I can purchase from PROTEUS, or elsewhere, the 218 keyboard encoder ROM on SDL's keyboard RC-board. One with SDL's original coding. As indicated above, I have the modified 16 which was supplied with the SDL Keyboard Customizer and Maintenance Kit (PROTEUS Cat. Item 4), I have tried calling Barry Watsman several times but got no answer. I assume that he has been supplying PROTEUS with the modified chip (218). I have been looking for wanting the 218 with original coding, 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 bad bug crept into the 218 and replacement of the bad 218 and replacing it with a source code, as in ZIBA and ZIBB. But knowing that a ROM can give me a feeling of insecurity. I need the assurance of a workable backup. Maybe a spare ZIBA would also be a prudent investment. I will call you about a week for what information you can give me on the availability of the ZIBA and ZIBB.

There is an interesting sidelight to the development of MYPEN. By assembling the objective code it was necessary to rewrite the assembler. The assembler language source for MYPEN is about 35K in length. MYPEN is itself about 4K in length, hence add this 5K for the job.

Letters published in the past few issues of the NEWS have expressed concern over the lack of continued support for the preventer, and I have had from time to time, since thought about writing a version of PENCIL-L in 8080 assembly language, the needs expressed in the letters provided the motivation to do something about it. After many years of translating and modifying the disassembled PENCIL-L version, I am now presenting this letter so that PENCIL-L is available 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 95K (a total of 253K bytes). The size of the program can now assemble virtually limited only by the size of the source code that can be written with the EDITOR in 52K RAM.

Sincerely yours,

Robert E. F. Bartels

P.O. Box 2248
Ann Arbor, MI 48106

[55 note: Proheus has an electronic parts supplier which can copy the 218 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 Proheus office.]

HELP! NEED TO DUMP DATA FROM MODEN TO OSBORNE/P. Kittle

**** NAILUTCHI SIDERIAH ****

Paul and Bev Kittle
Box 1286
Loma Linda, CA 92534

PROTEUS
1680 Woodside Road, Suite 219
Redwood City, CA 94061-3482

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 it would be nice to try out Richard Greenlaw's "disk航道.com" copy/m download from the old Solus 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 us to capture from my drives through Sol. None of the vendors I can find support my particular set-up, and 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, Mail merge, and SuperSort to manipulate citations from commercial data bases, so if this type of thing interests anyone, let me know. All copyright negotiations 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)
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 202 controller with 2, 8" drives running under CP/M 2.2.

I recently completed the SUPER PHANTOM ( VQL 5 2 Proteus News ) modification to my SOL and thank Fr. Thomas McGhee 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 SOLON on and thus turning the stack off, the routine as shown won't work. It turns SOLON off before attempting to relocate the stack. To fix this, the first four lines of the SOLON routine should be relocated to 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 SOLON 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 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 8010 Star system)
2. a version of Pencil which allows for non-printable control characters (my RS line printer will use underline, change fonts, etc.)
3. ...and many H of $.\n
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), cursor control for graphics and for word processing work (move this to type here?)

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 CMR, here I come.

Long live the SOL.

Keith R. Bettis