PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

Jan/Feb 1980
Vol. 3, # 1

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

Single Issue..$3 (US)

EDITORIAL

WHEW! Its 3:00 A.M. in the morning and I've just completed the final pasting together of my first issue of PROTEUS. All that remains is my first editorial. It's amazing to think of all the changes that have occured in my life and the field of micro-computers since I first bought my Sol20 kit three and one half years ago. But, one thing that hasn't changed is the plain and simple fact that the Sol20/Helios was the best of all the micros then and continues to be the best, in my opinion, now. WordWizard has yet to be duplicated by any other micro. As a programmers aid, PTDOS and the Helios operating system surpass any other system I have worked on, including the Corvus Disk.

As far as my philosophy goes, I am a networker. I believe in the dissemination of information to the needy at the smallest cost. I keep a disk full of names, addresses, phone numbers and areas of expertise of all the people I meet who seem willing to share their knowledge.

With this issue you may notice a change in emphasis. I do not plan on changing the direction PROTEUS has been going in, but, I do plan on emphasizing the sharing of resources. By resources I mean software, experiences, trouble-shooting procedures, repair stations and equipment. My dream would be for PROTEUS to be a central point for all to contact for those of you in need. I have even changed my data service for small businesses into a mail order service center for Sol/Helios/Processor Tech. equipment.

If you have a need, let us know what it is and we will do our best to find a way of filling it. In future issues I plan on including ideas on how to make a living off your computer, if you are not already. As Processor Tech. equipment owners, we need to support each other as much as possible. One way of doing this is to find all the P.T. owners out there. There are many owners out there who, believe it or not, do not know about us. If each of you would place an ad in one of your free advertiser papers we could blanket the U.S. in no time at all to try and reach these people. Local computer stores would also like to know about PROTEUS for the people who come into their stores looking for help. Many of these people bought the Sol for a purpose in their business 2/3 years ago and do not have any computer background and need us.

So PROTEUS can do the kind of thing it is doing, we also need for you to ask your fellow Sol users to subscribe to PROTEUS themselves instead of borrowing your copy. The extra dollars could help us buy all P.T. source codes and help us advertise in magazines in an effort to reach those who need us. It also just occured to me (I do my best thinking at 3:30 A.M.) that many schools (High Schools to be specific) bought Sols and then, when the computer hobbist/teacher left the school, the Sol got left in the closet. You might inquire of your local

Continued on back page.

Fixing Cassette BASIC's REM Statement
by Bob Werner

It irritated me that Processor Technology's Extended Cassette BASIC removed blanks between the keyword REM and the following remark. I often want spaces preceding a remark in order to indent it according to its import. I began putting an innocuous character, period, following REM. While this period allows me to position the text of the remark at will, it is still unsightly.

I disassembled BASIC for the usual reasons. In looking at the code, I discovered the reason for the missing blanks. The culprit is the keyword lookup routine. This routine is entered with a pointer to the unknown keyword on a program line, and another pointer to the beginning of the keyword table. A loop begins by bumping the keyword table pointer to the next letter. Then the program calls a subroutine to return the next non-blank character from the input line. This subroutine skims over any blanks and returns the first character it finds. The loop compares the characters. If they match, fine! the program returns to the top of the loop. If the characters don't match, the program checks to see if the end of the current keyword has been reached. If so, the keywords match and the program jumps off to subsequent routines. If it is not the end of a keyword, the program checks for an abbreviated command, and so on.

The blanks are missing in a REM statement because the loop calls the 'skim over blanks' subroutine even after the loop has found a perfect match for the keyword. You can solve the problem by rearranging the loop contents. The loop should first check if it has encountered the end of an entry in the keyword table. If not, only then should it call the 'skim over blanks' subroutine.

I have listed below the old and new code for the loop. I suggest using the following procedure to update your copy of Extended Cassette BASIC.

1. Put the BASIC cassette in your recorder and GET BASIC. Do not XEQ BASIC--you want to simply load it into memory at this time.

2. Use the Solos DUMP command to print the contents of the affected area:

```
     DUMP E0 EF
```

Verify the contents of memory against the 'old' listing below. If the contents match, fine. Proceed with the rest of the modification. If the contents do not match, I don't know what to tell you, other than don't continue trying to make this modification.

3. Use the Solos ENTER command to change the memory locations according to the 'new' listing.

Continued on page 2.

Continued from page 1.

4. You must also change the checksum for the BASIC program. When BASIC is first executed, a small program scans memory and creates a checksum of the entire BASIC interpreter. The computed checksum is compared with one recorded with the program. If the checksums do not match, the program alerts you with a message indicating what the checksum was found to be, and what it should have been. By rearranging the loop, you modify the checksum computed by the small program. You need to change the recorded checksum as well. So, again, use the DUMP command to display the recorded checksum:

```
        DUMP 3F81 3F82
```

This should display the values FE 9D, assuming you have already incorporated Processor Technology's fix for a FOR-NEXT problem that was published in ACCESS Vol. 2 No. 1. Again, do not proceed unless the values match.

5. Use the ENTER command to change the checksum bytes to 85 9D.

6. Place a fresh cassette in the recorder, record 15 seconds of leader, and save the modified BASIC program:

```
        SET TYPE 42
        SET XEQ 0
        SAVE BASIC 0 3F84
        SAVE BASIC 0 3F84
```

(I always record a program twice on a cassette, but that's up to you.)

7. At this point, you have the modified BASIC interpreter on cassette tape, and in memory. Check your modifications by executing BASIC:

```
        EX 0
```

Try typing in a small program of REM statements, with blanks following the REM. Then list the program. The blanks should still be there.

The 'Old' Listing

```
                    ...
00E0 13      LOOP INX  D    Point to keyword letter
00E1 CDCB02       CALL 2CB  Return next nonblank source letter
00E4 23           INX  H    Point to next source letter
00E5 47           MOV  B,A  Copy source letter
00E6 1A           LDAX D    Pick up keyword letter
00E7 B8           CMP  B    Same?
00E8 CAE000       JZ   LOOP Yes, do it again
00EB B7           ORA  A    No, end of keyword?
00EC FA4101       JM   GOTIT+1 Yes, don't bump HL
00EF 78           MOV  A,B  No, source letter to A
                    ...
```

The 'New' Listing

```
                    ...
00E0 13      LOOP INX  D    Point to keyword letter
00E1 1A           LDAX D    Pick up keyword letter
00E2 B7           ORA  A    End of keyword?
00E3 FA4001       JM   GOTIT Yes
00E6 47           MOV  B,A  Save keyword letter
00E7 CDCB02       CALL 2CB  Return next nonblank source letter
00EA 23           INX  H    Point to next source letter
00EB B8           CMP  B    Characters match?
00EC CAE000       JZ   LOOP Yes, loop
00EF 00           NOP       No, source letter already in A
                    ...
```

EPROM SERVICE ASSISTANCE

Fr. Thomas McGahee
Don Bosco Tech
202 Union Ave.
Paterson, N.J. 07502
August 23, 1979

Stan Sokolow
Proteus

Dear Stan,
On page 7 of the SOLUS NEWS, VOL 1 #5, there was a letter by Lewis Moseley Jr., at the end of which he suggested a few things that SOL users could use. I am pleased to announce that I am now in a position to make some of these suggestions a reality.

PROGRAMMING EPROMS: I will be happy to blast EPROMS from data contained on CUTS format tapes. I can do this for the 2708 (1K by 8) and 2716 (2K by 8 **5 VOLT ONLY**) Eproms. To avail yourself of this service, prepare a clean copy of the program on tape in CUTS format. Include on a sheet of paper the name of the program and its first and last memory locations. Include the necessary 2708/2716 EPROMS with your shipment TOGETHER WITH SUFFICIENT POSTAGE FOR THE RETURN OF YOUR TAPE AND EPROMS. Include also a return address label with your name and address neatly printed or typed on it (a 3 x 5 index card is acceptable). Your tape and EPROM(s) will be returned in the original packaging that I receive them in. Please make sure that the program to be copied is near the beginning of the tape so I don't waste lots of time trying to locate it! I will charge $5 for blasting each EPROM. If you have a used EPROM that needs to be erased, CLEARLY INDICATE THIS and include $1 for each EPROM that must be erased. For my part, I will verify each blasted EPROM against the tape copy. In the case of programs that require multiple EPROMS, I will afix a numbered label to each to identify which is which. It is a good idea to provide two copies of your program on the tape just in case one copy should turn out to be defective. I suggest padding any unused program space with FFs, which represents the unprogrammed state of the EPROM. I can also copy EPROMS at $5 per copy. Please include a note to the effect that the programs to be copied are your own, or are being copied with permission of the owner of any copyright that may be in effect. This is meant to be a service to SOL users, not a means for the illegal copying of copyrighted software!!! Follow all directions and I should be able to get your EPROMS back in the mail within a day or two.

I can also copy or transfer between the following media: CUTS format tapes at 1200 baud; 8" CP/M compatible disk in either single or double density SOFT SECTOR; 5 1/2" SOFT SECTOR CP/M compatible diskette (disk has a single sector hole);2708 EPROM; 2716 EPROM (5 volt type). I also hope within the near future to support 5 1/2" double density soft sector, but do not at present. There is a $5 fee for such media transfers, with you supplying all pertinent media, return postage, and return address label. Again, I require a note signed by you to the effect that you own the rights to the program or have obtained the necessary permission to have the program copied.

When asking for media transfers, please give sufficient information to allow me to do what you want (I cannot read your mind). [For instance, when transferring from Eprom to disk, what NAME and FILETYPE should be the program be assigned?] If at all possible, send media that contain ONLY the program(s) to be transferred. I assume no responsibility for possible damage that may occur to media; send decent media, not the most ragged, worn out, beaten-up cassette tape you happen to find lying around on the floor. NOTE:

Continued on page 3.

because of the time involved in transfers to and from tape, the $5 charge applies to each 24K that is transferred.

A rather extensive set of 8080 utilities have been written by myself and Bro. Al Roman. They will be offered on CUTS tape and on disk (in the disk formats mentioned above). This is a 2K program that assumes a memory mapped video display (such as a SOL or VDM). As supplied, it is customized for a standard SOL using SOLOS, but is easily modified for ANY 8080 computer. (its sole interface to the host computer hardware is a keyboard routine... this is currently set to access the keyboard directly using the port addresses for a standard SOL). A preliminary copy of this program and its documentation has been made available to Mr. Lewis Moseley Jr., who has agreed to evaluate it and send this evaluation in to PROTEUS. A brief description of its utilities is included here for thosewho might be interested: It is a menu-driven program. When waiting for input, it displays a list of the available commands that are chosen by typing a single key. It allows dumping of memory in ASCII HEX, with EACH byte preceded by its address. It allows the entry of HEX, and the finding of hex-code sequences up to 215 bytes long. It allows filling sections of memory with any code. It allows moving of blocks of data either up or down. It allows the entry of ASCII directly into memory, including ALL control codes, and can even append an optional terminator of your choice. It can search for the occurrences of any ASCII string (including control codes) up to 432 characters in length. Note that the HEX and ASCII searches result in the found code being shown on the screen IN CONTEXT, as well as returning the address of the match. Further, the limits of the memory to be searched can be changed using a built-in command. Pages of memory can be displayed. A special paging mode allows the user to 'flip' through memory pages without having to enter the addresses manually. Entering an address forces that page to be displayed. A RELOCATOR similar to that which OSUDAR and MAGUIRE used for relocating the ALS/8 is also included. This relocator allows simple entry of all necessay parameters. As an example, a step-by-step relocation of the entire set of utilities is detailed, showing how the relocator can be used. This relocator can be used to relocate any program so long as you know where the data areas are located.

The above-mentioned utilities bear the name **THE MODIFIER II**. It is offered on CUTS tape for $10, on CP/M disk for $13, including documentation. As a service to interested users, an assembly listing is also available. It is $10 on CUTS cassette in Processor Tech Assembler format (written on the ALS/8), including a hardcopy of the listing. If ordered at the same time as THE MODIFIER, the program and assembler listing will be placed on the same tape and the total cost will be $17 (since I save on postage and tape costs, so do you). An assembly listing on disk is also available, written in CP/M Assebler format. Ordered seperately, it is $13. If I put the program and assembler on the same disk, you get both for $23, including a hardcopy listing. When ordering, please specify tape or disk, and if disk, whether 8" soft sector or 5 1/2 " soft sector. (If you want double density 8", please add $2 per disk, as the double-density disks cost me more).

As you can see from the prices, I am not out to make money, but rather to provide a service to SOL users (and any others who wish to take advantage of my offers). Every effort will be made to provide fast service. Please help by including all pertinent information with your order.

Sincerely yours,

Fr. Thomas McGahee

Fr. Thomas McGahee S.D.B.

P.S. Make all checks/money orders out to
Don Bosco Technical High School,
but send all orders/correspondence
directly to Fr. Thomas McGahee.

SOL & HELIOS REPAIR

Dear PROTEUS,

For the last year I have free-lanced out of the "Computer Demo Room" (BYTE Shop) in San Rafael. I am now working out of my home. I have all of the diagnostics and necessary tools to repair the Sol and align the Helios.

My hourly rates are $25.00 plus expenses and .21 cents per mile. For a flat rate of $60.00 I will pick up and deliver anywhere in the greater San Francisco Bay Area.

Thank you

Albert A Bonelli III
P.O. Box 294
Petaluma, Ca. 94952
(707)762-9514

NOTES NOTES NOTES NOTES NOTES NOTES NOTES NOTES

Quill floppy diskettes seem to not have a good polish in the recording area.
Georgia Magnetics have a good polish.

NKRA INFORMATION

PROMS	P.T. PART #
U37 AND U33	726107
U39	726108
U40	726109
U56	726111

THE NKRA DELAY LINE IS 704002...ONE SOURCE WAS :

ENGINEERED COMPONENTS CO. (Part # STTLDM-409 Active delay, 5 tap)

NOTE TO THE NOVICES

How do you write the "END" file on a tape using the Solos "SAVE" command?

Save a file named "END" with identical start and end addresses. This creates a file of length =1 byte. BASIC text files write file of length zero as an end mark, but Solos recognizes no end-of-file mark. Your convention is your decision, but the file "END" is commonly used.

INDEX to Extended Cassette BASIC Manual

Continued on page 11.

NORTH STAR ERROR FIX

Dear Stan,

OOPS, I goofed. In my program for loading North Star Files (PROTEUS, v. 2, no. $, July-August, 1979), line 77 now reads

```
LXI H,5200H
```

It should read

```
LXI B,5200H.
```

I hope that this has not been misleading to anybody; if so I apologize.

This has been an instructive error for me, and it came about in the following way. The program developed over a period of time. As usual in such circumstances, it possessed certain inelegancies, which I tried to remove for publication. I give your readers the following distillations from my experience.

1. If a program works, leave it the way it is, even if it has warts. Better warts than bugs.

2. If you do decide on cosmetic surgery, then you should not only make sure the program assembles without error messages, but also make sure it runs.

3. Remember that as you sew, so shall you rip.

Sincerely,

THINKER TOY 2D

Dear Stan:

I noticed in your issue (May-Jun) that you were planning to buy a Thinker Toy 2D system.

After a goodly wait, I have a two drive system on my Sol, along with CP/M.

If you reset the Sol with CP/M in RAM, you are not able to jump back into CCP because of a hardware error in Thinker Toy...unless you first change location E0F4 hex to 00 hex. I learned this from a technician at Thinker Toy over the phone.

Using Thinker Toy with Sol results in a disk with considerably less than the advertised 600K bytes when used with CP/M which I also purchased. You get 77 tracks of 26 sectors, each with 256 bytes, or 512,512 bytes.

The customized BIOS for Sol when received although made by Lifeboat Associates for the Sol, unexpectedly did not provide for control P, printer output. I was able to add a software fix for this, as shown by this letter.

A good point for Thinker Toy is that the ROM boot is at E000 to start things going, which permits anyone with a northstar disk system to keep it in, and use either.

Sincerely
Bernard Plotkin, Mgr
Rendale Hotel
3120 Collins Ave.
Miami Beach, Florida 33140

ROY HEYBROCK

CERTIFIED FINANCIAL PLANNER
2516 Lakeshore Dr.
~~1803 E. Wendover Avenue~~
Greensboro, North Carolina 2740~~5~~7

Res: 299-7698

Bus: 272-4591

Dear Stan:

First of all I want to thank you for your efforts in putting togather the PROTEUS / NEWS. Everybody always likes to read about how good their equipment is and hopes that everyone else thinks it is the greatest also. I am no exception. I own the whole system , SOL, HELIOS and a Centronics printer. I have had my system for about a year and it has done everything I have needed to do. Sometimes I had to try several times but mainly I was trying to get too complicated. Most programs could be done a lot easier than I was trying to do it.

Please send a list of prices and items for the PTC liquidation if it is not too late. Enclosed is my self addressed stamped envelope.

I noticed your NOVICE-TO-NOVICE article. I know i'm a novice and really feel dumb about all the things my SOL can do and I don't even know what all it can do. First of all I don't understand why or how for that matter anyone uses assembly language. If I did maybe I could find some use for the peek and poke and other such statements in BASIC. I know they are used by PET but only because I read listings in Magizines about it. I bought Aosborne & Associates "8080A/8085 Assembly Language Programming" but couldn't find any reason to,say,store a " H " in some location in memory.Some people tell me it runs a lot faster but the trade off to learn how to write ASS'Y doesn't seam to be worth it. If someone could help in this area I'm sure that I am not alone.
I have been writing one long program mostly but have started breaking my programs into groups of shorter programs lately. This is done by storing data in a data file on disk and acessing that data with each sub program. To do this you can have options for the operator and the program will then "XEQ XXX". The XXX is the name of your branch to program. You can then read back the necessery data, make changes or whatever and then "XEQ YYY", where YYY is the name of your original or main program. I find this very usefull when I want to make changes for only part of the data entry information. A specific example is used by the Word Wizard whenthey set up for the printer. They set up the printer by using a specific pragram to SET OF = PRINT instead having to change the program or ask the operator which printer you want to use for this output. This example may seem academic to most of us who are lucky enough to have one printer let alone worry about which one to use. It works nifty for the days date however and anytime you find yourself cussing because you always seem to have to enter the same imformation over and over.

Sincerly

Roy Heybrock

Roy

"PLANNING FOR FINANCIAL INDEPENDENCE"

PROTEUS NEEDED EVEN MORE

Dear Stan,

This is the first time anyone from this area has written to Proteus as far as I know. We have an active Sol user group here in Fargo, with many machine combinations represented. We have Sol/Helius, Sol/NorthStar, and ,of course, Sol/cassette.

We have had many of the same problems in dealing with Processor Tech that other users have had, and apparently, their attitude fianally got the best of them. In a conversation with the people at PerSci, I found that PTCO is bankrupt. I trust this is the truth, since calls there result in a "line disconnected" response.

Most of us feel it really won't present a problem, since we have not had a problem we can't fix in over two years. All of our problems in the Helios for instance have either been in the power supply or in the failure of the head positioning lamp. In the Sol, most problems have been in the memory boards.

I have CPM/CBASIC2 as well as PTCO software, and some good utility programs from Computer Port in Texas. One of our local users has written a text formatter for use with files prepared by EDIT or EDIT 3.0. We have both cassette and disk versions, and we have special drivers for the NEC Spinwriter built into the program.

I personally hope the downfall of PTCO does not cause a similar problem for Proteus, since it is even more important to have a strong user network to share all information now.

If any of the Proteus readers care to have any of our information, I hope they will write me at the address below. Keep up the good work, I'm looking forward to the next issue.

Sincerely

Bill Pederson
415 Forest Ave. North
Fargo, ND 58102

QUESTIONAIRE LACKING

Dear PROTEUS

I believe your questionaire would have probably been more enlightening if it had asked, "What external devices do you have/want?"

Because there are likely many like me (particularly in isolated places) who have not bought mass storage because of rapid improvement in the field and the lack of reliability in some discs when compared to tape.

It strikes me that the large mainframe users practice of a combined tape and disc system is the way to go!

B.W.Littlejohn
Box 58, RR1
Chimney Valley Estates
Williams Lake, B.C. V2G 2P1

EDITOR: How about it, anyone want to take this on and submit your ideas to PROTEUS? I've gotten to where I never even think of tape for backups. I've had good results and experience with Helios and North Star disc drives. My experience with the discs themselves has been reasonably good. I do make several backups of all my programs.

COLOR GRAPHICS BOARD SOUGHT

Dear PROTEUS,

Enclosed is my check for my 1980 dues. Keep up the good work, I have gotten quite a bit of information out of my copies of PROTEUS.

I have been looking for a color graphics board that is compatible with my Sol, but have been a little leary as to which one to buy. Maybe you can help me with this.

Thank you

Donald F. Siebenrock
RD#2
Clearfield, Pa. 16830

EDITOR: Stan or I have no direct information on color graphics boards. Anyone out there able to help us?

NEVADA COBOL PRICES SLASHED!

A new product, NEVADA COBOL, is now available to translate source language programs into machine language on 8080/z80/8085 microprocessors. Designed specifically for small businesses using microprocessors, NEVADA COBOL is simple, fast and easy to use. Standard Features include:

- Random access file support
- Sequential files both fixed and variable length
- Debugging capability
- Copy statement
- Data types: Character string, 16-bit binary and packed decimal.
- 18-digit accuracy
- Hexidecimal non-numerick literals
- Powerful editing with English language error messages
- Interactive ACCEPT/DISPLAY
- Subset of ANSI-74

The high-performance compiler generates fast in-line machine language object programs at a rate up to 650 lines/minute on an 8080-CPU. Currently operating under PTDOS and requires a minimum of 32K.

Outstanding price/performance and delivery is immediate. Want a closer look? The NEVADA COBOL Programmers Reference Manual is available for $10 and the Diskette is $65. California residents please add sales tax.

Ellis Computing
1480 17th Ave.
San Francisco, CA 94122

EDITOR: This package used to sell for $25 and $275 respectively. Will anyone who has tried it review it?

MODIFICATIONS TO PTC ECBASIC

Enclosed is my PROTEUS membership application for 1980 and a check for $63: $18 for dues and $45 for Proteus item P13, source and object code for Extended Cassette BASIC. Also enclosed is source code for a set of changes, I have made to Extended Cassette BASIC, and an example run of BASIC after the changes were made.

I hope that Proteus is able to complete the deal with PTC for PTC source code. However, if it should fall through, I suggest that we disassemble the code ourselves. I have all ready disassembled Extended Cassette BASIC into 165 pages of source code, and I have annotated about 65 of those pages. Legally we may not be able to exchange such source code. But we should be allowed to write articles about what we have found; i.e., codeword/token/routine address tables, format of variables and strings in storage, flow charts, etc.

I have just completed a disc operating system which may be of use to some members. Your first reaction may well be, "Not another one!". I wrote it because I wanted to run existing software with as little change as possible, and with all of the functions of SOLOS intact. Therefore, I simply extended SOLOS by adding two new unit numbers, two new commands and one new SET command to add two single density, soft sectored, mini-floppy disc drives. With some pride, I call my extension to SOLOS Bolton's disc Operating System or BOS - I've always wanted to be in charge of or boss the machine rather than at the mercy of its shortcomings; but being in charge is a lot more work, I've found, than tagging along. Back to BOS. BOS has some hooks into SOLOS, so the SOLOS EPROM must be rewritten to use BOS. At the other end, BOS talks to the disc drives through calls to BIOS for CP/M. BIOS and diagnostic routines are in about 2K of EPROM at D000. BOS is in 2K of EPROM at D800. All variables for BIOS and BOS are stuffed into the system RAM at C000. However, there is still room for a good sized stack there as well. Since the listing alone for BOS is 30 pages, I am not going to send it in unless someone is interested in using it.

My drives are Shugart model SA400. The two drives and power supply are in a custom cabinet 17" wide, 9" deep and 5" high so that it fits nicely on top of the SOL with the monitor on top of the drive cabinet. I mounted the drive power supply regulators outside the cabinet on the back so that a fan is not required in the drive cabinet. The disc controller is the Versafloppy by S.D. Systems. On that board, I found the Wait State Generator would not work with the SOL. By cutting the existing trace to U18 pin 8 and routing the DRQ output (pin 38 of FD1771B-1) to that pin, I provided the same function in software by inspecting the state of bit 6 on input port X3 with no loss in speed. My firmware is in 2716s on the Cromemco 32K Bytesaver board. I have had no problems at all with it. Programming the 2716s (INTEL) is very easy. Furthermore, the 2708 Personality Module is very easily changed to accept a 2716: bypass R5; cut traces between feedthroughs 21 and C0, and 19 and +12; put jumper between feedthroughs C0 and 19. The 2716 goes in socket U1. The changes provide +5v to pin 21 (Vpp) and connects the NOT page C0 input to pin 19 (A10).

Sincerely,

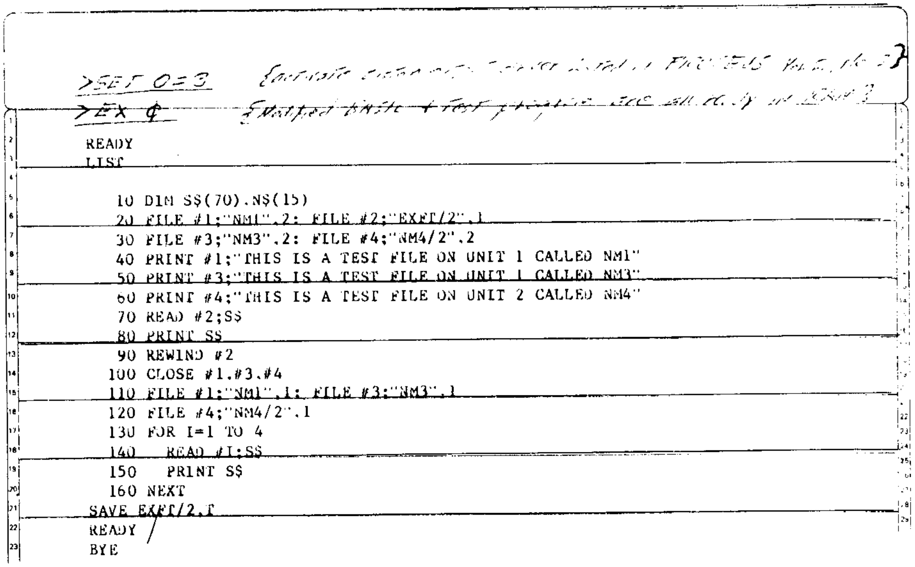
Ed Bolton

Ed Bolton
4253 Moore St.
LA,CA 90066
(213) 391-9998

```
0000           0010 * Modifications to PTC Extended Cassette BASIC
0000           0020 *  by E. W. Bolton
0000           0030 *  12/31/79
0000           0040 *
0000           0050 **** Change No. 1 ****
0000           0060 *
0000           0070 * 1] Eliminate LET from the program listing
0000           0080 * 2] Eliminate blanks from the exponent of
0000           0090 *    numbers in exponential notation when
0000           0100 *    writing to files ONLY.
0000           0110 *
0000           0120 * Side effect: Access granted variable is not set
0000           0130 *               by FILE statement, nearly ignored.
0000           0140 *
0000           0150       ORG 24DH
024D FE 98     0160       CPI 98H    ;If the token is for LET,
024F C3 AA 12  0170       JMP PLIST  ; skip it.
0252           0180 *
0252           0190       ORG 0F27H
0F27 CD B2 12  0200       CALL PPRNT ;Set to print blanks
0F2A           0210 *
0F2A           0220       ORG 136CH
136C CD B7 12  0230       CALL PPFIL ;Set to suppress blanks
136F           0240 *
136F           0250       ORG 1883H
1883 CD B2 12  0260       CALL PPRNT ;Set to print blanks
1886           0270 *
1886           0280 * Mods to FILE routine to make room for patches
1886           0290       ORG 12A4H
12A4 CD AB 0D  0300       CALL 0DABH ;Scan over access granted variable to
12A7           0310 *                 next statement; i.e., ignore it.
12A7 C3 F6 13  0320       JMP 13F6H  ;Put header into FCB & go to next statement
12AA           0330 * Room here now for patches
12AA 3E 20     0340 PLIST MVI A,20H  ;IF the last token was not for LET,
12AC C2 58 02  0350       JNZ 258H   ; output its keyword
12AF C3 98 02  0360       JMP 298H   ; ELSE skip it.
12B2 3E 20     0370 PPRNT MVI A,20H  ;Set to print blanks
12B4 C3 B8 12  0380       JMP PPFIL+1
12B7 AF        0390 PPFIL XRA A      ;Set to delete blanks
12B8 32 94 24  0400       STA 2494H
12BB C3 13 22  0410       JMP 2213H  ;Output a number
12BE           0420 * 3 bytes left in this area.
12BE           0430 *
12BE           0440 **** Change No. 2 ****
12BE           0450 *
12BE           0460 * This change integrates BASIC and BOS (Bolton's Disc
12BE           0470 * Operating System) to provide file access to floppy disc
12BE           0480 * files as well as cassette files.  The only change for
12BE           0490 * block access is to allow unit no. up to 4:
12BE           0500 *    Unit No.   Device
12BE           0510 *      1          floppy disc drive no. 1
12BE           0520 *      2             "     "    "    " 2
12BE           0530 *      3          cassette drive no. 1
12BE           0540 *      4             "       "    " 2 (not allowed for byte access)
12BE           0550 *
12BE           0560 * Byte access changes are:
12BE           0570 * 1] When calling FOPEN, in SOLOS, reg. E=access requested
12BE           0580 * 2] When calling any byte access routine, HL is expected to
12BE           0590 *    be pointing at the header.  With only one exception,
12BE           0600 *    this was all ready provided by BASIC.
12BE           0610 * 3] BASIC no longer prints any "PREPARE TAPE ..." messages
12BE           0620 *    nor waits before opening files.  They would be a waste
12BE           0630 *    of time when working with discs.
12BE           0640 * 4] BASIC now allows up to four files to be opened.
12BE           0650 *    (BOS constrains this to four DISC files OR
12BE           0660 *     two DISC files and one CASSETTE file on tape unit 1)
12BE           0670 *
12BE           0680 * File Control Blocks, FCBs, are stored at 15D7-15FE.
12BE           0690 * An FF byte at 15FF signifies their end (2 FCBs).
12BE           0700 * From 1602-163E, the "PREPARE TAPE ..." messages are
12BE           0710 * stored.  By using the message space for FCBs, more
```

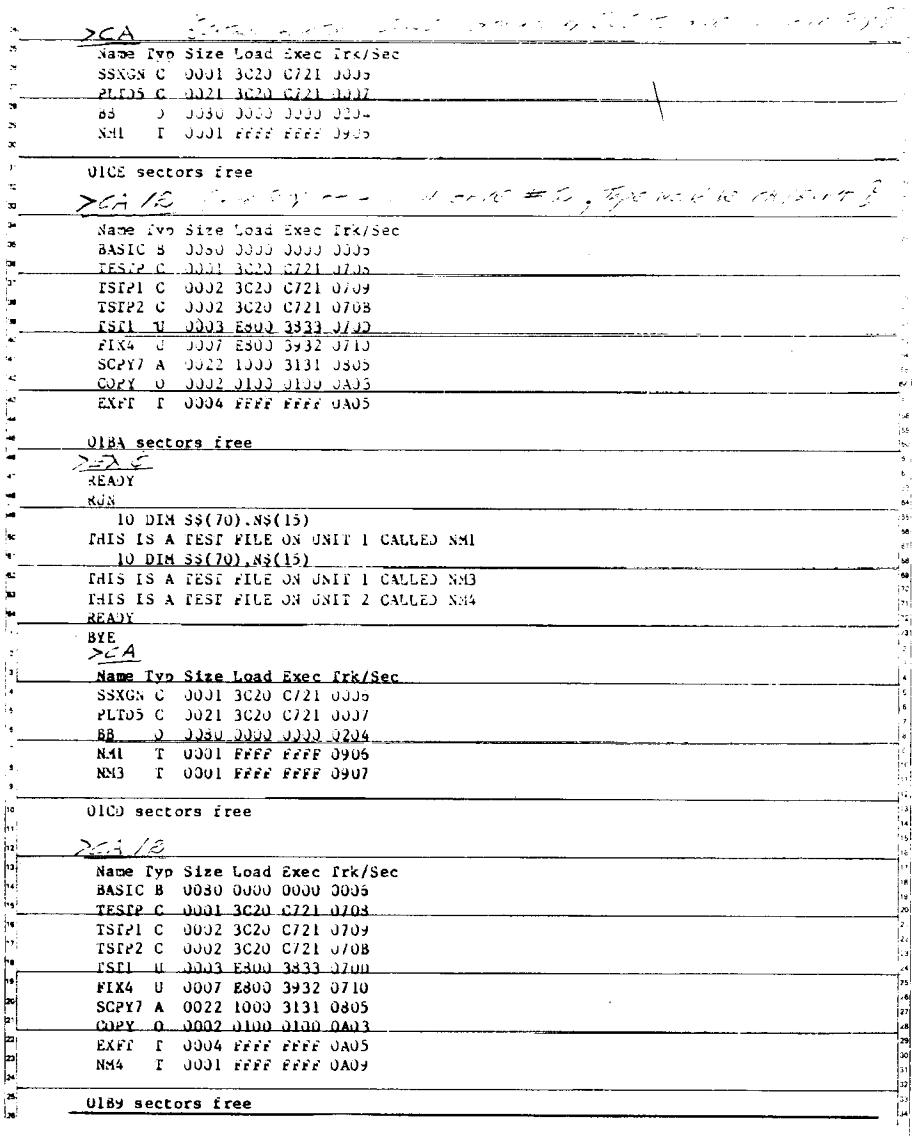
```
12BE          0720 * files may be opened.  However, a variable indicating
12BE          0730 * the address of the current FCB at 1600-1601 must be
12BE          0740 * moved.  The next few orders do that.
12BE          0750 *
12BE          0760 * Initialize 2 new FCBs to closed.
12BE          0770      ORG 15FFH
15FF 00       0780      DB 0
1600          0790      ORG 1613H
1613 00       0800      DB 0
1614          0810 * Relocate the end of FCBs byte.
1614          0820      ORG 1627H
1627 FF       0830      DB 0FFH
1628          0840 * Relocate the addr of current FCB to 1628-1629
1628          0850 * (future patch area remains at 162A-1640)
1628.         0860 ORG 970H
0970 28 16    0870 DW 1628H ;In SAVE routine
0972          0880 ORG 0A20H
0A20 28 16    0890 DW 1628H ;In XEQ.APPEND.GET routine
0A22          0900 ORG 13C6H
13C6 28 16    0910 DW 1628H ;ln locate FCB routine
13C8          0920 ORG 13DDH
13DD 28 16    0930 DW 1628H ; "
13DF          0940 ORG 13E4H
13E4 28 16    0950 DW 1628H ; "
13E6          0960 ORG 13EEH
13EE 28 16    0970 DW 1628H ;ln move FCB to header routine
13F0          0980 ORG 13F9H
13F9 28 16    0990 DW 1628H ;ln move header to FCB routine
13FB          1000 *
13FB          1010 ORG 144EH
144E 04       1020 DB 4      ;Allow unit numbers up to 4
144F          1030 *
144F          1040 * The following mods the REWIND routine to delete the messages
144F          1050 * and to set reg. E=access requested when calling FOPEN.
144F          1060 * Up to 1493, the file has been closed (to SOLOS).
144F          1070      ORG 1493H
1493 21 C4 15 1080      LXI H,15C4H
1496 7E       1090      MOV A,M       ;A=unit #
1497 23       1100      INX H
1498 5E       1110      MOV E,M       ;E=access requested
1499 23       1120      INX H
149A 23       1130      INX H         ;HL points to header
149B C3 BC 14 1140      JMP 14BCH     ;There we call FOPEN
149E          1150 * (future patch area available from here through 14BB)
149E          1160 *
149E          1170 * The following eliminates messages & E=access requested
149E          1180 * for BASIC's main byte file opening routine.
149E          1190      ORG 156BH
156B FE 03    1200      CPI 3
156D D2 FC 29 1210      JNC 29FCH ;AC error if access requested >2
1570 B7       1220      ORA A
1571 CA FC 29 1230      JZ 29FCH  ; or =0
1574 32 C5 15 1240      STA 15C5H
1577 5F       1250      MOV E,A     ;E=access requested
1578 C3 AB 15 1260      JMP 15ABH   ;Finish opening the file there
157B          1270 * (future patch area from 1581 through 15AA)
157B          1280 *
157B          1290 * Make HL point to name in FCB & A=unit no.
157B          1300 * when closing files of a crashed program.
157B          1310      ORG 32H
0032 CD 7B 15 1320      CALL CRSFX
0035          1330 *
0035          1340      ORG 157BH
157B 23       1350 CRSFX INX H
157C 7E       1360      MOV A,M      ;A=unit no.
157D 11 03 00 1370      LXI D,3      ;Set up to point HL at name in FCB
1580 C9       1380      RET
1581          1390 *
```

```
1581          1400 **** Change No. 3 ****
1581          1410 *
1581          1420 * Steamline the startup of BASIC so no questions are ask
1581          1430 * and all facilities are provided.  Also change the
1581          1440 * copyright message to a modified by me message to keep
1581          1450 * track of my various modified versions.
1581          1460 *
1581          1470    ORG 3D35H
3D35 2A 0A 09 1480    LHLD 90AH
3D38 C3 57 3D 1490    JMP 3D57H ;Use all of memory
3D3B          1500    ORG 3D5DH
3D5D C3 CF 3D 1510    JMP 3DCFH ;Keep all functions
3D60          1520 *
3D60          1530    ORG 3EC5H
3EC5 20 20    1540    DW '  '   ;"  modified by E. W. Bolton 12/30/79""
3EC7 20 6D    1550    DW 'm '
3EC9 6F 64    1560    DW 'do'
3ECB 69 66    1570    DW 'fi'
3ECD 69 65    1580    DW 'ei'
3ECF 64 20    1590    DW ' d'
3ED1 62 79    1600    DW 'yb'
3ED3 20 45    1610    DW 'E '
3ED5 2E 20    1620    DW ' .'
3ED7 57 2E    1630    DW '.W'
3ED9 20 42    1640    DW 'B '
3EDB 6F 6C    1650    DW 'lo'
3EDD 74 6F    1660    DW 'ot'
3EDF 6E 20    1670    DW ' n'
3EE1 31 32    1680    DW '21'
3EE3 2F 33    1690    DW '3/'
3EE5 30 2F    1700    DW '/0'
3EE7 37 39    1710    DW '97'
3EE9 22       1720,   DB '"'
3EEA          1730 *
3EEA          1740 * Update the checksum byte
3EEA          1750     ORG 3F81H
3F81 75 A4    1760     DW 0A475H
```



```
>SET O=3    {[illegible]}
>EX 0       {Modified BASIC + test program [illegible]}
READY
LIST

   10 DIM S$(70),N$(15)
   20 FILE #1;"NM1",2: FILE #2;"EXFT/2",1
   30 FILE #3;"NM3",2: FILE #4;"NM4/2",2
   40 PRINT #1;"THIS IS A TEST FILE ON UNIT 1 CALLED NM1"
   50 PRINT #3;"THIS IS A TEST FILE ON UNIT 1 CALLED NM3"
   60 PRINT #4;"THIS IS A TEST FILE ON UNIT 2 CALLED NM4"
   70 READ #2;S$
   80 PRINT S$
   90 REWIND #2
  100 CLOSE #1,#3,#4
  110 FILE #1;"NM1",1: FILE #3;"NM3",1
  120 FILE #4;"NM4/2",1
  130 FOR I=1 TO 4
  140   READ #I;S$
  150   PRINT S$
  160 NEXT
SAVE EXFT/2,T
READY
BYE
```



```
>CA   [illegible]
 Name Typ Size Load Exec Trk/Sec
 SSXGN C  0001 3C20 C721 0005
 PLT05 C  0021 3C20 C721 0007
 BB    0  0030 0000 0000 0204
 NM1   T  0001 FFFF FFFF 0905

0ICE sectors free
>CA /B   [illegible] #2, [illegible]
 Name Typ Size Load Exec Trk/Sec
 BASIC B  0030 0000 0000 0005
 TESTP C  0001 3C20 C721 0705
 TSTP1 C  0002 3C20 C721 0709
 TSTP2 C  0002 3C20 C721 070B
 TST1  U  0003 E800 3333 0700
 FIX4  U  0007 E800 3932 0710
 SCPY7 A  0022 1000 3131 0805
 COPY  0  0002 0100 0100 0A03
 EXFT  T  0004 FFFF FFFF 0A05

01BA sectors free
>EX S
READY
RUN
   10 DIM S$(70),N$(15)
THIS IS A TEST FILE ON UNIT 1 CALLED NM1
   10 DIM S$(70),N$(15)
THIS IS A TEST FILE ON UNIT 1 CALLED NM3
THIS IS A TEST FILE ON UNIT 2 CALLED NM4
READY
BYE
>CA
 Name Typ Size Load Exec Trk/Sec
 SSXGN C  0001 3C20 C721 0005
 PLT05 C  0021 3C20 C721 0007
 BB    0  0030 0000 0000 0204
 NM1   T  0001 FFFF FFFF 0906
 NM3   T  0001 FFFF FFFF 0907

01CD sectors free

>CA /B
 Name Typ Size Load Exec Trk/Sec
 BASIC B  0030 0000 0000 0005
 TESTP C  0001 3C20 C721 0703
 TSTP1 C  0002 3C20 C721 0709
 TSTP2 C  0002 3C20 C721 070B
 TST1  U  0003 E800 3833 0700
 FIX4  U  0007 E800 3932 0710
 SCPY7 A  0022 1000 3131 0805
 COPY  0  0002 0100 0100 0A03
 EXFT  T  0004 FFFF FFFF 0A05
 NM4   T  0001 FFFF FFFF 0A09

01B9 sectors free
```

PROGRESS REPORT FROM YOUR PUBLISHER

Source code purchase: No news to report. It's just a slow process beyond my control. I'll keep you informed.

Soft-sectored controller for Helios: Tarbell DD controller is out--they aren't interested in PerSci drives; likewise, ThinkerToys; Micromation is all talk and no action as far as I am concerned; Delta Products has incompatibilities with Sol; a tech from DataSpeed is coming to me to check-out his DD controller. I still have others to check.

Future items: I have Helios versions of PILOT, VULCAN database system, and Nevada COBOL for review.

Stan Sokolow

MAIL ORDER SERVICE CENTER OPENS

After talking to many of the PROTEUS members on the telephone I have decided to start a mail order service center for Sol equipment owners.

The one overriding concern that many of you have is the worry of what do you do if your system goes bad. Well, I am currently in touch with four ex-Processor Tech trained repair people who are willing to work on any Processor Tech equipment, including the notorious 16k memory boards. (Data delays are on order and expected in 3-6 months)

My position would be one of a representative of the owner of the equipment to be repaired. I would only send your equipment to someone who I felt was qualified as a repair person and charged reasonable rates. I personally have two Sol/Helios systems and have had to seek repair from these people myself. I have found placed that do repair to the PERSCI disks but not to the tolerance needed desperately for Helios. (Did you know that most PERSCI repair people adjust the PERSCI to operate in a "soft sectored" configuration. The Helios uses 32 hard sectored disks and the tolerances are much smaller, so small, in fact that the PERSCI can be operating perfectly in another system but fail when connected to a HELIOS and most PERSCI repair people do not have a Sol20 to test the HELIOS.)

So, be wary of repair people who don't even have Sol equipment.

As far as parts is concerned I am trying and am compiling a list of all used and new Processor Tech. equipment that was sold or given to people who worked with P.T. in lew of wages during the end. Many of these people are willing, for a cost of course, to sell the extra equipment they have laying around the house.

So, if something like your Helios goes down, you call me. I will try and help you troubleshoot what is wrong (so you dont pay for nothing) and I will try and give you an estimated price as to the cost to repair your item. You send to me (after packaging the equipment nicely) your piece of equipment. You include a check for the estimated amount of the repair bill plus $50 made out to me. The $50 will be my fee I will then find the repair person who can get to your piece of equipment as soon as possible and do the best repair.

I will then personally transport your piece of equipment to the repair person and pick it up when finished. I will then TEST the item on my system to make sure it works. That way any problem can be solved without you having to wait and pay shipping costs unnecessarily.

All the repair people I will deal with will have Sols to test their repairs. I will then carefully pack the item in your shipping case and ship it back to you COD for the shipping costs. Any refund due you will also be sent. If any major repair must be done on your system I will see to it that you are notified and approve the new repair bill. Also, on items found to be ok, you will have to pay the repair people the cost of troubleshooting. (A reasonable rate when you compare it to a company near here that charges $90 just to look at your system, bad or good.)

I will also be willing to act as an intermediary for any of you who need memory, etc. and want to buy it from me. The costs will be retail costs but at least you know the equipment works with Sol equipment and the warranty is good.

This way you have someone who is your representative and someone who will take an interest in your equipment and its operation. I will also stay on top of the parts situation so that any unresonable costs can be challanged by me face to face with the repair people rather than by telephone or mail.

Please feel free to call me if you have any questions at 707-446-0417 or drop me a line at 131 Highland Ave, Vacaville, Ca. 95688

Note: This is a function being provided by Tony's Data Service and not an official PROTEUS project.

SOLO FOR CP/M USERS

Dear Stan,

I am enclosing a copy of the manual for a program I have written called SOLO. This program is aimed at SOL users who have CP/M disk systems. In brief, this program loads under CP/M and then relocates itself above the user's present BIOS, and then patches the JUMP TABLE in the user's BIOS so that CONSTAT, CONIN, and CONOUT then reference the routines included in SOLO. SOLO splits the screen intc two sections. The upper section is fourteen lines in length, and represents the OUTPUT section. The last two screen lines comprise a COMMAND LINE, representing the INPUT section.

When a carriage return is input, the COMMAND LINE is made available a character at a time to the calling routine. Until this terminating carriage return is hit, the entire 128 character COMMAND LINE can be edited using the facilities built into SOLO. These utilities include delete left, delete right, clear the entire line, go to beginning of line, go to end of line, move cursor left, move cursor right, insert text (shifting remaining text to right), two tab modes (tabe code and spacing to next tab position). The program also enables the user to transfer output into the COMMAND LINE: this allows a simple way to edit lines in BASIC and when using a text editor. To maintain full compatibility with all other programs, the user can switch back and forth between the new COMMAND LINE mode and the old character-at-a-time input mode. This is accomplished by typing CTRL/X. Control codes used by CP/M are handled accordingly, but the ESCAPE key can be used to cause any code to be entered as pure code rather than having the SOLO program interpret it for you. This allows full flexibility in using control codes when running programs such as BASIC under CP/M.

Using SOLO makes using any program easier, since it provides a complete set of editing facilities that remain the same no matter what program you are using! The unique TRANSFER command allows you to edit lines that have already been entered; in MICROSOFT BASIC, this provides a way to change the number of a line without retyping the entire line, among other things.

Extensive use is made of the special keys available on a SOL, and yet alternate codes have also been made available that are the same as those used by the ELECTRIC PENCIL. This was done so that learning the editing codes would be a simple matter, with a minimum of "unlearning". The program SOLO is being made available to SOL owners in the following formats: CUTS 1200 baud tape, suitable for use on ANY SOL RUNNING CP/M; 8" IBM single density disk, soft sector; 8" double density, soft sector; 5 1/4" single density disk, soft sector. Micropolis and other disk systems should order the CUTS tape (instructions are included detailing how to get the program from tape onto disk). Cost is $10 for tape and single densitv disk versions. Add $3 for double density disk. If you supply your own media, then all versions are $8. All versions include manual.

I hope that readers who know of SOL users running CP/M will pass on this information. I an trying to do my part to help support the SOL user. I cannot afford to advertise in the usual magazines, due to my low prices which really only cover my operating overhead. I rely upon this newsletter and word of mouth to inform people of the existence of this support software.

Sincerely yours,

Fr. Thomas McGahee

Fr. Thomas McGahee S.D.B.

PROGRAM NAME: SOLO **CATEGORY:** OPERATING SYSTEM

DESCRIPTION: Program self-relocates itself above your present CP/M BIOS and patches JUMP TABLE so new CONSTAT, CONIN, and CONOUT routines are available. New routines allow FULL EDITING including cursor left, cursor right, go to beginning of line, go to end of line, delete left, delete right, INSERT, and TRANSFER. COMMAND LINE may be edited with great ease, and it also supports a SUBMIT mode in which many command "lines" may be written as a single COMMAND LINE. To allow complete compatibility with ALL programs running under CP/M, ESCAPE key allows direct entry of ANY control code. CTRL/X allows switching between the COMMAND LINE mode with its built-in editing facilities, and a character-at-a-time input routine that is similar to what most CP/M users currently use.

MINIMUM HARDWARE REQUIRED: 8080/Z80 system set up for CP/M. SOL or computer with VDM is assumed in standard version. (Inquire about non-standard video versions). Memory mapped video at CC00 is standard, but other addresses supported.

SOFTWARE REQUIRED: CP/M. (All I/O done using direct port access, so will work with all SOL type computers, and easily patched for any hardware configuration... assumes no software other than CP/M).

RESTRICTIONS: Assumes user has at least 2K of memory above BIOS. This is easily ensured with any system having at least 18K of memory. A list of offsets for all system sizes is included to aid those who might have to perform a MOVCPM.

DOCUMENTATION: Manual explaining theory of operation and each edit operation is included. Includes notes on using the facilities to edit programs written in MICROSOFT DISK BASIC. A handy reference sheet giving a quick overview of all the edit facilities is included.

MEDIA: CUTS Tape, 1200 Baud, suitable for ANY disk system, with instructions. DISK: 8" single density, soft sector ("standard" CP/M). 8" double density, soft sector (MICROMATION DOUBLER, etc). 5 1/4" single density, soft sector (VISTA, VERSAFLOPPY, etc).

DATE CURRENT VERSION WAS RELEASED: September, 1979

WARRANTY: Defective media replaced if returned within 14 days. If not satisfied, return within 14 days for refund (minus postage).

PRICE: Tape=$10. Single density disk=$10. Double density 8" disk=$13. If YOU provide the media, cost is $8 (any version).

ORDER FROM: Fr. Thomas McGahee, 202 Union Ave., Paterson, N.J. 07502. Send check or money order with order. Shipped First Class mail, normally within 24 hrs after receipt of order. CUSTOMIZED versions may take longer.

REMARKS: CUSTOMIZATION available for non SOL users who have a 16 X 64 memory mapped video device. For customization, include a listing of your current CONSTAT and CONIN routines, and the address of your Video display memory. If your video board includes a hardware scrolling port, include the address of this port. Customized key codes are also available. Customized programs cost an additional $10. 24 x 80 versions are possible; if interested, send a stamped, self addressed envelope.

```
10 PRINT CHR$(11)
20 REM              PROGRAM 'CHGBAS'
30 REM      PROGRAM WILL CHANGE THE BASE OF ANY NUMBER INPUT
40 REM      AS DESCRIBED IN THE PRINTED INSTRUCTIONS FOLLOWING
50 REM
60 REM       COPYRIGHT BY JACK W. RANDOLPH
70 REM       FEBRUARY, 1979
80 REM       RANDOLPH & ASSOCIATES, INC.
90 REM       586 SHADES CREST ROAD
100 REM      BIRMINGHAM, AL 35226   PH 205-822-2339
110 REM
120 !"THE FOLLOWING PROGRAM WILL CONVERT ANY NUMBER INPUT"
130 !"WHETHER BASE 8,10 OR 16 (OCTAL, DECIMAL OR HEX)."
140 !"IF DECIMAL NUMBERS ARE INPUT, THE HEX AND OCTAL"
150 !"EQUIVALENT WILL BE GIVEN..IF HEX IS INPUT, THE DECIMAL"
160 !"AND OCTAL EQUIVALENT WILL BE GIVEN, ETC."
170 !
180 !"AN 'H' FOR 'HEX' AND AN 'O' FOR 'OCTAL' MUST FOLLOW"
190 !"HEX AND OCTAL VALUES....THIS IS THE ONLY REQUIREMENT"
200 !\!
210 INPUT "INPUT NUMBER YOU WISH TO CONVERT: ",X$
220 A$="ABCDEF"
230 M$=X$(LEN(X$))\REM..LAST CHARACTER
240 REM ASSUME DECIMAL IF NOT 'O' OR 'H'
250 IF M$="O" THEN 320
260 IF M$="H" THEN 690
270 IF M$<="9" THEN 280\GOTO 290
280 IF M$>="0" THEN 870
290 !"NUMBER INPUT MUST END IN 'O' OR 'H' IF NOT"
300 !"A DECIMAL VALUE...GOING BACK TO BEGINNING! "
310 GOTO 210
320 REM....OCTAL CONVERSION
330 FOR I=1 TO LEN(X$)-1
340 IF X$(I,I)>="0" THEN 360
350 GOTO 370
360 IF X$(I,I)<="7" THEN 400
370 !"NO DECIMAL NUMBERS PERMITTED LESS THAN ZERO"
380 !"OR GREATER THAN SEVEN, IF OCTAL NUMBER"
390 GOTO 210
400 NEXT
410 J=8
420 S=0
430 GOSUB 440\C=1\GOSUB 770\END
440 I=LEN(X$)
450 I=I-1
460 IF I=0 THEN 670
470 S=S+1
480 IF X$(I,I)<="9" THEN 580
490 M2=0
500 FOR K=1TO6
510 IF X$(I,I)<>A$(K,K) THEN 530
520 X1=K+9\GOTO 590
530 M2=M2+1
540 NEXT K
550 IF M2<>6 THEN 590
560 !"INVALID HEX CHARACTER ",X$(I,I)
570 GOTO 210
580 X1=VAL(X$(I,I))
590 IF S<>1 THEN 610
600 S1=X1\GOTO 450
610 X2=1
620 FOR M=1 TO S-1
630 X2=X2*J
640 NEXT M
650 S1=S1+X2*X1
660 GOTO 450
670 PRINT "DECIMAL VALUE IS ",S1
680 RETURN
690 FOR I=1 TO LEN(X$)-1
700 IF X$(I,I) >="0" THEN 710\GOTO 560
710 IF X$(I,I)<="F" THEN 720\GOTO 560
720 NEXT
730 X=0\J=16\GOSUB 440
740 C=2\J=8
750 GOSUB 770
760 INPUT"FINISHED? ",X1$\IF LEN(X1$)=0 THEN 120
770 IF C=1 THEN 810
780 IF C=2 THEN 840
790 REM C=1 IS DEC TO HEX CONVERSION
800 REM C=2 IS DEC TO OCTAL CONVERSION
810 J=16\GOSUB 950
820 PRINT "HEX VALUE IS: ",N$
830 RETURN
840 J=8\GOSUB 950
850 PRINT "OCTAL VALUE IS: ",N$
860 RETURN
870 FOR I=1TO LEN(X$)-1
880 IF X$(I,I)>"9" THEN 560
890 IF X$(I,I)<"0" THEN 560
900 NEXT
910 S1=VAL(X$)
920 C=1\GOSUB 770
930 C=2\GOSUB 770
940 INPUT "FINISHED? ",X1$\IF LEN(X1$)=0 THEN 120
950 N$=""
960 R=S1
970 IF R>0 THEN GOSUB 1040 ELSE RETURN
980 IF L>9 THEN 1000
990 K$=STR$(INT(L))\K$=K$(2)\GOTO 1010
1000 M=L-9\K$=A$(M,M)
1010 N$=K$+N$
1020 R=K
1030 GOTO 970
1040 K=INT(R/J)
1050 L=R-K*J
1060 RETURN
```

Continued from page 4.

ERROR MESSAGES

AC	Access error	A3-4
AM	Argument error	A3-2
BS	Bad syntax	A3-1
CA	Cannot append	A3-4
CL	Close error	A3-4
CS	Control stack error	A3-3
DD	Double definition	A3-2
DI	Direct execution error	A3-2
DM	Dimension error	A3-2
DZ	Divide by zero	A3-3
FD	Format definition/file declaration	A3-1
FM	Format error	A3-3
FO	Field overflow	A3-3
FP	Floating point overflow	A3-3R
IN	Input error	A3-1R
IS	Internal stack error	A3-2
LL	Line length	A3-1
LN	Line number reference	A3-2
MD	Matrix dimension error	A3-5
MS	Matrix singular error	A3-5R
NC	Not continuable	A3-2R
NI	Function deleted	A3-3R
NP	No program	A3-2
OB	Out of bounds	A3-3
OP	Open error	A3-4
RD	Read error	A3-4
SO	Storage overflow	A3-3
TY	Variable type error	A3-2
UD	Undimensioned matrix	A3-5R
WT	Write error	A3-4

Stan,

I have discovered a potential source of trouble for those hobbyists that have a Sol with an SSM 8K memory card or a WAMECO MEM1 8K memory card. The problem is that both the data drivers on the Sol and these memory cards are enabled simultaneously. This bus conflict should cause concern if it were only 20ns in duration, but in this case it is several clock cycles long!

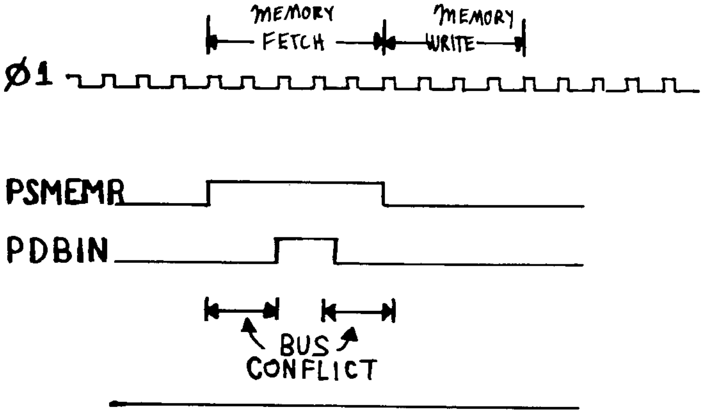
I was alerted to this area by Howard Fullmer, who explained that there was a 20ns bus conflict in the Sol, when used with good memory cards. But the problem could be much worse, depending on your memory cards. The two memory cards previously mentioned do not decode PDBIN, they only decode SMEMR. Therefore, when doing a read from memory, they are enabled for the entire instruction cycle. Unfortunately, the Sol drivers are enabled for the entire memroy read instruction cycle except while PDBIN is active. This causes a bus conflict of several cycles (see drawing). In addition, there is another bus conflict of about 20ns that can occur even if you memory cards decode PDBIN. This is because the same signal (PDBIN) is used to turn on one set of drivers and turn off the other set of drivers. Since there are several levels of gates (with various and somewhat unpredictable PROP delays) between the two sets of drivers, there is about a 20ns bus conflict.

My system appeared to run properly with these bus conflicts, but, they can cause nasty power supply spikes and probably don't help the MTBF of your drivers.

SUGGESTION:

1. Determine if your memory cards decode PDBIN, If they don't, then add it.
2. Cut pint 10 of U45 (74LS04) on your Sol and connect pin 5 of U47 (74LS10) to pint 5 of U107 (SWO). This will disable the Sol output data drivers for the entire instruction cycle of a memory read or I/O read.

Gordon Wilson



An Instruction Decoder for the ALS8 Simulator

by Joe Maguire
P.O. Box 3742 DT
Anchorage, AK 99510

Of all the debugging tools in my software arsenal, the ALS8 Simulator remains one of the best. It is one of the few which can be used to trace program flow through ROMS. (It uses no RST instructions) However, it was frustrating not to be able to "see" the instruction mnemonics corresponding to the machine code being displayed. A session with the Simulator required constant referal to the 8080 instruction set.

Realizing that there were nine blank columns remaining on the video display gave me the idea to include a modified instruction decoder as part of the simulator package. Since the instruction mnemonic must fit within the nine spaces, some abbreviation was required. The addresses of JMPs and CALLs as well as the data values of the LXIs have been omitted. These are readily determined from the display itself. The remaining instructions are displayed normally.

The routine can be assembled to reside in any area of RAM with about 700 bytes available including within the ALS8 RAM area itself. (In my own version, relocated to address zero, I put the routine at 0300H) Only one patch to the ALS8 is required.

Note: This routine is intended to be used with a "RAM" version of ALS8. It could be used with the "ROM" version if the required software were added. (intercept the RET instruction at address F73B from the stack)

A sample display is given below:

```
C004  00000 00  00 00 00 00  00 00 21  0000  C3 C1 C9  JMP
C1C9  00000 00  00 00 00 00  00 00 21  CBFF  31 FF CB  LXI SP
C1CC  00000 00  00 00 00 00  00 00 21  CBFF  3A 07 C8  LDA C807
C1CF  00000 00  00 00 00 00  00 00 21  CBFD  F5 00 00  PUSH A
C1D0  00011 00  00 00 00 00  00 00 21  CBFD  AF 00 00  XRA A
C1D1  00011 00  00 00 00 00  00 00 21  CBFD  32 07 C8  STA C807
C1D4  00011 00  00 00 00 00  00 00 21  CBFB  CD C2 F1  CALL
C2F1  00011 00  00 00 00 00  00 00 21  CBF9  CD C2 F9  CALL
C2F9  00011 00  0A 00 00 00  00 00 21  CBF9  06 0A 00  MVI B,0A
C2FB  00011 00  0A 00 00 00  00 00 21  CBF7  CD C0 19  CALL
```

```
0000                  0010 *
0000                  0020 *      ALS8 filename "SIMID"
0000                  0030 *
0000                  0040 *  ALS8 SIMULATOR INSTRUCTION DECODER
0000                  0050 *  Written by Joe Maguire, Oct 1979,
0000                  0060 *  for ALS8 relocated to address 0000.
0000                  0070 *  Portions of the disassembler code
0000                  0080 *  obtained from the CP/M library.
0000                  0090 *
0000                  0100 *  (ALS8 is a software product of
0000                  0110 *   Processor Technology Corp.
0000                  0120 *   Addresses in () reference the
0000                  0130 *   original code location)
0000                  0140 *
0000                  0150 *  Note: The ALS8 must be patched at
0000                  0160 *    address 2730H (F730) to jump to
0000                  0170 *    the origin of this routine.
0000                  0180 *    Example:  2730  C3 00 C9
0000                  0190 *
0000                  0200 ASCII  EQU    156FH (E56F)
0000                  0210 COUT   EQU    00CFH (D0CF) Rev. B
0000                  0220 LAST   EQU    2733H (F733)
0000                  0230 INSTS  EQU    014CH (D14C) Rev. B
0000                  0240 PSW    EQU    6
0000                  0250 *
```

```
0000                0260            ORG     0C900H
C900                0270 *                           ORIGINALLY THE LAST CALL
C900 CD 33 27       0280 ENTRY      CALL    LAST  -  BY THE SIMULATOR BEFORE
C903                0290 *                           PROCESSING THE NEXT LINE
C903 06 20          0300            MVI     B,' '
C905 CD BC CB       0310            CALL    CO
C908 CD BC CB       0320            CALL    CO         THE CURRENT SIMULATOR
C90B 21 4C 01       0330            LXI     H,INSTS -  Instruction stored
C90E 22 51 CA       0340            SHLD    TEMP       AT THIS LOCATION
C911 CD 5C CB       0350            CALL    RDBYT
C914 57             0360            MOV     D,A
C915 21 53 CA       0370            LXI     H,TABLE
C918 01 11 00       0380            LXI     B,11H
C91B                0390 *
C91B BE             0400 D1         CMP     M
C91C CA 46 CA       0410            JZ      TG1
C91F 23             0420            INX     H
C920 0D             0430            DCR     C
C921 C2 1B C9       0440            JNZ     D1
C924 0E 0A          0450            MVI     C,0AH
C926 BE             0460            CMP     M
C927                0470 *
C927 BE             0480 D2         CMP     M
C928 CA 35 CA       0490            JZ      TG2
C92B 23             0500            INX     H
C92C 0D             0510            DCR     C
C92D C2 27 C9       0520            JNZ     D2
C930 0E 06          0530            MVI     C,6
C932                0540 *
C932 BE             0550 D3         CMP     M
C933 CA 14 CA       0560            JZ      TG3
C936 23             0570            INX     H
C937 0D             0580            DCR     C
C938 C2 32 C9       0590            JNZ     D3
C93B E6 C0          0600            ANI     0C0H
C93D FE 40          0610            CPI     40H
C93F CA FD C9       0620            JZ      MG0
C942 FE 80          0630            CPI     80H
C944 CA EE C9       0640            JZ      MG1
C947 7A             0650            MOV     A,D
C948 E6 C7          0660            ANI     0C7H
C94A D6 04          0670            SUI     4
C94C CA E2 C9       0680            JZ      MG2
C94F 3D             0690            DCR     A
C950 CA DC C9       0700            JZ      MG3
C953 3D             0710            DCR     A
C954 CA C8 C9       0720            JZ      MG4
C957 7A             0730            MOV     A,D
C958 E6 C0          0740            ANI     0C0H
C95A CA AD C9       0750            JZ      MG5
C95D 7A             0760            MOV     A,D
C95E E6 C7          0770            ANI     0C7H
C960 D6 C0          0780            SUI     0C0H
C962 CA A5 C9       0790            JZ      MG6
C965 D6 02          0800            SUI     2
C967 CA 9D C9       0810            JZ      MG7
C96A D6 02          0820            SUI     2
C96C CA 95 C9       0830            JZ      MG8
C96F D6 03          0840            SUI     3
C971 CA 8C C9       0850            JZ      MG9
C974 7A             0860            MOV     A,D
C975 E6 07          0870            ANI     7
C977 4F             0880            MOV     C,A
C978 21 43 CB       0890            LXI     H,PPOP-1
C97B 09             0900            DAD     B
C97C CD 7C CB       0910            CALL    PRINT
C97F CD 90 CB       0920            CALL    XTRAC
C982 FE 06          0930            CPI     6
C984 C2 EB C9       0940            JNZ     D6
C987 06 41          0950            MVI     B,'A'
```

```
C989 C3 BC CB       0960            JMP     CO
C98C                0970 *
C98C 21 40 CB       0980 MG9        LXI     H,PRST
C98F CD 7C CB       0990            CALL    PRINT
C992 C3 6F 15       1000            JMP     ASCII
C995                1010 *
C995 06 43          1020 MG8        MVI     B,'C'
C997 CD BC CB       1030            CALL    CO
C99A C3 97 CB       1040            JMP     CCPRN
C99D                1050 *
C99D 06 4A          1060 MG7        MVI     B,'J'
C99F CD BC CB       1070            CALL    CO
C9A2 C3 97 CB       1080            JMP     CCPRN
C9A5                1090 *
C9A5 06 52          1100 MG6        MVI     B,'R'
C9A7 CD BC CB       1110            CALL    CO
C9AA C3 97 CB       1120            JMP     CCPRN
C9AD                1130 *
C9AD 21 28 CB       1140 MG5        LXI     H,PLXI
C9B0 7A             1150            MOV     A,D
C9B1 E6 0F          1160            ANI     0FH
C9B3 3D             1170            DCR     A
C9B4 CA C2 C9       1180            JZ      MG51
C9B7 FE 04          1190            CPI     4
C9B9 DA BE C9       1200            JC      D4
C9BC D6 05          1210            SUI     5
C9BE                1220 *
C9BE 87             1230 D4         ADD     A
C9BF 87             1240            ADD     A
C9C0 4F             1250            MOV     C,A
C9C1 09             1260            DAD     B
C9C2 CD 7C CB       1270 MG51       CALL    PRINT
C9C5 C3 AB CB       1280            JMP     RPPRN
C9C8                1290 *
C9C8 21 24 CB       1300 MG4        LXI     H,PMVI
C9CB CD 7C CB       1310            CALL    PRINT
C9CE CD 90 CB       1320            CALL    XTRAC
C9D1 CD 65 CB       1330            CALL    RGPRN
C9D4 06 2C          1340            MVI     B,','
C9D6 CD BC CB       1350            CALL    CO
C9D9 C3 40 CA       1360            JMP     D8
C9DC                1370 *
C9DC 21 20 CB       1380 MG3        LXI     H,PDCR
C9DF C3 E5 C9       1390            JMP     D5
C9E2                1400 *
C9E2 21 1C CB       1410 MG2        LXI     H,PINR
C9E5 CD 7C CB       1420 D5         CALL    PRINT
C9E8 CD 90 CB       1430            CALL    XTRAC
C9EB C3 65 CB       1440 D6         JMP     RGPRN
C9EE                1450 *
C9EE 7A             1460 MG1        MOV     A,D
C9EF E6 38          1470            ANI     38H
C9F1 0F             1480            RRC
C9F2 4F             1490            MOV     C,A
C9F3 21 FC CA       1500            LXI     H,PADD
C9F6 09             1510            DAD     B
C9F7 CD 7C CB       1520            CALL    PRINT
C9FA C3 0E CA       1530            JMP     D9
C9FD                1540 *
C9FD 21 F8 CA       1550 MG0        LXI     H,PMOV
CA00 CD 7C CB       1560            CALL    PRINT
CA03 CD 90 CB       1570            CALL    XTRAC
CA06 CD 65 CB       1580            CALL    RGPRN
CA09 06 2C          1590            MVI     B,','
CA0B CD BC CB       1600            CALL    CO
CA0E                1610 *
CA0E 7A             1620 D9         MOV     A,D
CA0F E6 07          1630            ANI     7
CA11 C3 65 CB       1640            JMP     RGPRN
CA14                1650 *
```

```
CA14 79              1660 TG3     MOV     A,C
CA15 87              1670         ADD     A
CA16 87              1680         ADD     A
CA17 4F              1690         MOV     C,A
CA18 F5              1700         PUSH    PSW
CA19 21 DC CA        1710         LXI     H,TAB3-4
CA1C Ø9              1720         DAD     B
CA1D CD 7C CB        1730         CALL    PRINT
CA2Ø F1              1740         POP     PSW
CA21 FE ØC           1750         CPI     12
CA23 D8              1760         RC
CA24 FE 14           1770         CPI     2Ø
CA26 DØ              1780         RNC
CA27                 1790 *
CA27 CD 5C CB        1800 D7      CALL    RDBYT
CA2A 57              1810         MOV     D,A
CA2B CD 5C CB        1820         CALL    RDBYT
CA2E CD 6F 15        1830         CALL    ASCII
CA31 7A              1840         MOV     A,D
CA32 C3 6F 15        1850         JMP     ASCII
CA35                 1860 *
CA35 79              1870 TG2     MOV     A,C
CA36 87              1880         ADD     A
CA37 87              1890         ADD     A
CA38 4F              1900         MOV     C,A
CA39 21 B4 CA        1910         LXI     H,TAB2-4
CA3C Ø9              1920         DAD     B
CA3D CD 7C CB        1930         CALL    PRINT
CA4Ø                 1940 *
CA4Ø CD 5C CB        1950 D8      CALL    RDBYT
CA43 C3 6F 15        1960         JMP     ASCII
CA46                 1970 *
CA46 79              1980 TG1     MOV     A,C
CA47 87              1990         ADD     A
CA48 87              2000         ADD     A
CA49 4F              2010         MOV     C,A
CA4A 21 70 CA        2020         LXI     H,TAB1-4
CA4D Ø9              2030         DAD     B
CA4E C3 7C CB        2040         JMP     PRINT
CA51                 2050 *
CA51                 2060 TEMP    DS      2
CA53                 2070 *
CA53 ØØ              2080 TABLE   DB      Ø
CA54 Ø7              2090         DB      7
CA55 ØF              2100         DB      ØFH
CA56 17              2110         DB      17H
CA57 1F              2120         DB      1FH
CA58 27              2130         DB      27H
CA59 2F              2140         DB      2FH
CA5A 37              2150         DB      37H
CA5B 3F              2160         DB      3FH
CA5C 76              2170         DB      76H
CA5D C9              2180         DB      ØC9H
CA5E E3              2190         DB      ØE3H
CA5F E9              2200         DB      ØE9H
CA6Ø EB              2210         DB      ØEBH
CA61 F3              2220         DB      ØF3H
CA62 F9              2230         DB      ØF9H
CA63 FB              2240         DB      ØFBH
CA64 C6              2250         DB      ØC6H
CA65 CE              2260         DB      ØCEH
CA66 D3              2270         DB      ØD3H
CA67 D6              2280         DB      ØD6H
CA68 DB              2290         DB      ØDBH
CA69 DE              2300         DB      ØDEH
CA6A E6              2310         DB      ØE6H
CA6B EE              2320         DB      ØEEH
CA6C F6              2330         DB      ØF6H
CA6D FE              2340         DB      ØFEH
CA6E 22              2350         DB      22H
CA6F 2A              2360         DB      2AH
CA7Ø 32              2370         DB      32H
```

```
CA71 3A              2380         DB      3AH
CA72 C3              2390         DB      ØC3H
CA73 CD              2400         DB      ØCDH
CA74                 2410 *
CA74 45 49           2420 TAB1    ASC     'EI'
CA76 20 20           2430         DW      2020H
CA78 53 50 48 4C     2440         ASC     'SPHLDI  XCHG'
     44 49 20 20
     58 43 48 47
CA84 50 43 48 4C     2450         ASC     'PCHLXTHLRET HLT '
     58 54 48 4C
     52 45 54 20
     48 4C 54 20
CA94 43 4D 43 20     2460         ASC     'CMC STC CMA DAA '
     53 54 43 20
     43 4D 41 20
     44 41 41 20
CAA4 52 41 52 20     2470         ASC     'RAR RAL RRC RLC NOP '
     52 41 4C 20
     52 52 43 20
     52 4C 43 20
     4E 4F 50 20
CAB8                 2480 *
CAB8 43 50 49 20     2490 TAB2    ASC     'CPI ORI XRI ANI '
     4F 52 49 20
     58 52 49 20
     41 4E 49 20
CAC8 53 42 49 20     2500         ASC     'SBI IN  SUI OUT '
     49 4E 20 20
     53 55 49 20
     4F 55 54 20
CAD8 41 43 49 20     2510         ASC     'ACI ADI '
     41 44 49 20
CAEØ                 2520 *
CAEØ 43 41 4C 4C     2530 TAB3    ASC     'CALLJMP LDA '
     4A 4D 50 20
     4C 44 41 20
CAEC 53 54 41 20     2540         ASC     'STA LHLDSHLD'
     4C 48 4C 44
     53 48 4C 44
CAF8                 2550 *
CAF8 4D 4F 56 20     2560 PMOV    ASC     'MOV '
CAFC                 2570 *
CAFC 41 44 44 20     2580 PADD    ASC     'ADD ADC SUB SBB '
     41 44 43 20
     53 55 42 20
     53 42 42 20
CBØC 41 4E 41 20     2590         ASC     'ANA XRA ORA CMP '
     58 52 41 20
     4F 52 41 20
     43 4D 50 20
CB1C                 2600 *
CB1C 49 4E 52 20     2610 PINR    ASC     'INR '
CB2Ø 44 43 52 20     2620 PDCR    ASC     'DCR '
CB24 4D 56 49 20     2630 PMVI    ASC     'MVI '
CB28 4C 58 49 20     2640 PLXI    ASC     'LXI STAXINX DAD '
     53 54 41 58
     49 4E 58 20
     44 41 44 20
CB38 4C 44 41 58     2650         ASC     'LDAXDCX '
     44 43 58 20
CB4Ø 52 53 54 20     2660 PRST    ASC     'RST '
CB44 50 4F 50 20     2670 PPOP    ASC     'POP PUSH'
     50 55 53 48
CB4C 4E 5A 5A 20     2680 CCODE   ASC     'NZZ NCC POPEP M '
     4E 43 43 20
     50 4F 50 45
     50 20 4D 20
CB5C                 2690 *
CB5C 2A 51 CA        2700 RDBYT   LHLD    TEMP    Read a byte
CB5F 7E              2710         MOV     A,M
CB6Ø 23              2720         INX     H
```

```
CB61 22 51 CA          2730        SHLD   TEMP
CB64 C9                2740        RET
CB65                   2750 *
CB65 3C                2760 RGPRN  INR    A       Register Print
CB66 E6 Ø7             2770        ANI    7
CB68 FE Ø6             2780        CPI    6
CB6A DA 6F CB          2790        JC     RGP1
CB6D C6 Ø3             2800        ADI    3
CB6F FE Ø5             2810 RGP1   CPI    5
CB71 DA 76 CB          2820        JC     RGP2
CB74 C6 Ø2             2830        ADI    2
CB76 C6 41             2840 RGP2   ADI    41H
CB78 47                2850        MOV    B,A
CB79 C3 BC CB          2860        JMP    CO
CB7C                   2870 *
CB7C ØE Ø4             2880 PRINT  MVI    C,4
CB7E 46                2890 P1     MOV    B,M
CB7F CD BC CB          2900        CALL   CO
CB82 23                2910        INX    H
CB83 ØD                2920        DCR    C
CB84 C2 7E CB          2930        JNZ    P1
CB87 78                2940        MOV    A,B
CB88 FE 2Ø             2950        CPI    ' '
CB8A C8                2960        RZ
CB8B Ø6 2Ø             2970        MVI    B,' '
CB8D C3 BC CB          2980        JMP    CO
CB9Ø                   2990 *
CB9Ø 7A                3ØØØ XTRAC  MOV    A,D     Extract
CB91 E6 38             3Ø1Ø        ANI    38H
CB93 ØF                3Ø2Ø        RRC
CB94 ØF                3Ø3Ø        RRC
CB95 ØF                3Ø4Ø        RRC
CB96 C9                3Ø5Ø        RET
CB97                   3Ø6Ø *
CB97 CD 9Ø CB          3Ø7Ø CCPRN  CALL   XTRAC  Condition Code
CB9A 87                3Ø8Ø        ADD    A
CB9B Ø6 ØØ             3Ø9Ø        MVI    B,Ø
CB9D 4F                31ØØ        MOV    C,A
CB9E 21 4C CB          311Ø        LXI    H,CCODE
CBA1 Ø9                312Ø        DAD    B
CBA2 46                313Ø        MOV    B,M
CBA3 CD BC CB          314Ø        CALL   CO
CBA6 23                315Ø        INX    H
CBA7 46                316Ø        MOV    B,M
CBA8 C3 BC CB          317Ø        JMP    CO
CBAB                   318Ø *
CBAB CD 9Ø CB          319Ø RPPRN  CALL   XTRAC  Reg. Pair
CBAE E6 Ø6             32ØØ        ANI    6
CBBØ FE Ø6             321Ø        CPI    6
CBB2 C2 65 CB          322Ø        JNZ    RGPRN
CBB5 Ø6 53             323Ø        MVI    B,'S'
CBB7 CD BC CB          324Ø        CALL   CO
CBBA Ø6 5Ø             325Ø        MVI    B,'P'
CBBC                   326Ø *
CBBC C3 CF ØØ          327Ø CO     JMP    COUT   CHARACTER OUTPUT
CBBF                   328Ø *
```

OUT OF '77' AND '78' ISSUES

PROTEUS has run out of 1977 and 1978 back issues. So, please do not order any back issues from these years. We do, however, have plenty of back issues for 1979. By the way, look for a "BEST OF SOLUS/PROTEUS NEWS" later this year.

North Star Library Forming

We have one person at present working on some of the software we have for the North Star/Sol20 pair. The work consists of sorting through 70 or so disks and finding the gems and seeing to it they are bug free and documented.

We plan to operate the North Star library similiar to the Helios Library and will come out with specifics in our next newsletter.

Those of you who have software can submit your programs as per our policy on copyrights by sending them to Tony Severa, 131 Highland Ave, Vacaville, Ca. 95688. I will return your disks as soon as possible with credit for purchases of our library.

There's no reason the North Star library cannot be as active as the Cassette / Helios Software Library. Costs will be $10 per disk if you submit an "approved" program and $25 if you do not submit an approved program.

NORTH STAR HELP NEEDED

Hi Tony,

Enclosed is a check to cover 1980 dues. Next, a big THANK-YOU! for your tips on locating a used North Star Horizon as per our phone conversation. A follow up to one of your suggestions landed me just what I was looking for. I didn't receive it yet, but I'd like to run my Sol software on the Horizon.

1. Do you or anyone you know, know how I can do this, and get my Sol software on disk?

2. I want to use the Horizon Controller board and drives for my Sol too, what kind of North Star format disk will I need for Basic (Sol Basic) and do lyou lhave a Sol (Solos) monitor I could load to make the North Star think its a Sol?

3. Do you have North Star formatted disks with Sol Software (to run on Sol) already on it. I'm especially interested in Gamepac 1 and Music (Software Tech) and would like also to be able to run these on my horizon.

I've ordered a DataSouth adapter for LA35/36 Decwriters to upgrade from 300 baud to 1200 baud. If you want, I'll evaluate it and send you a camera ready report on it and how easy (? difficult) it was to connect.

Thanks

Ron Genova
601 Friedensburg Road
Pennside
Reading, Pennsylvania 19606

EDITOR: 1. I do have some Sol software on my North Star disk. What I did was: 1. Initialize North Star D.O.S. 2. Reset the Sol with the UpperCase and Repeat keys. 3. Load by cassette Sol software to memory 2D00. (North Star D.O.S. resides from 2000 to 2CFF.) 4. Type EXEC 2028 (This will put you back into N.S. D.O.S.) 5. Create your file name and length of file type 1 with the starting address needed for the Sol. 6. WD 2D00 and # of blocks.
This is how I have done it with North Star in the past. If someone has a better idea please send it in and I will publish it.

Proteus Cassette Software Library
Operating Policies and Procedures

PURPOSE: The Proteus Cassette Software Library exists to serve the Proteus membership. The library accepts contributions of software from members, and makes it available to the membership at an attractive price. As such, it is a dynamic entity; it must grow to survive. As an incentive to contributors, the library software is offered at a discount to those who have contributed programs or data files.

PURCHASES: Current library cassettes are priced at $16.ØØ each. This price includes a quality cassette in a hard plastic case, packaging in a protective envelope, and postage. All cassettes will be sent via U.S. mail. All orders must be prepaid. Checks and money orders are accepted; send cash at your own risk. Payment must be made in U.S. dollars; foriegn purchasers add $1.ØØ for air mail shipment.

Certain of the older library cassettes are overstocked, and are being closed out at a discount while supplies last. These were listed in the newsletter, V2,#4. Future production runs of these cassette will be sold at the price which is then current.

On current library cassettes, a credit of $1Ø.ØØ is allowed for the donation of acceptable software to the library. Thus, the net price is $8.ØØ with discount. On the closeout cassettes, the price is lower, and therefore the credit is also lower.

COPYRIGHTS: To the extent possible, all Proteus Cassette Software library programs are Copyrighted. This is done solely to prevent commercial exploitation. Programs which are shown as copyrighted by the individual authors, or otherwise explicitly marked as having restricted reproduction rights, may not be reprinted or copied without the permission of the author. All other programs may be copied for any NON-COMMERCIAL purpose, provided that proper credit is given to Proteus. For information concerning reproduction for commercial purposes, contact the Proteus main office at 169Ø Woodside Rd., Suite 219, Redwood City CA 94Ø61.

Thus, except for the restricted programs, a club or local group can buy a single copy and share it among its members. (We know you do it anyway, we're just making it legal.) To make it even easier, a copy of TCOPY, the CUTS tape copy/verify utility by Lewis Moseley, Jr., will be included on most current and new cassettes.

CONTRIBUTIONS: The library is constantly seeking new software. However, we are able to accept software for distribution only if it is in the public domain, or with the permission of the author or copyright holder. Programs copied from magazines and books are generally not acceptable, since they are copyrighted by the author or publisher. Similarly, propriatary programs can not be accepted. Remember, though, that you cannot copyright an idea, or an algorithym; just the specific way of expressing it. So, it is possible to use the ideas of a published program to write a similar one of your own. The contribution of an original program or data base earns you a purchase discount credit.

Certain magazines, notably Dr. Dobbs Journal and Recreational Computing, give blanket permission to use their programs, so long as proper credit is given. Programs from these sources are acceptable, provided that you have entered the necesary credit lines. In this case, you are donating your time and effort in keying in the program, rather than the program itself; you still get a discount credit.

In summary, then, for your program to be accepted you must send a signed statement that you know the program to be in the public domain OR that the program is your original creation and that you give the library permission to distribute it OR a letter from the author or copyright holder giving the library permission to distribute OR a program from a publication which gives permission to reprint. Unless one of these conditions are met, the program can not be used, and a credit will not be issued. Note that the library does not insist that your rights to a program be signed over to us; all we ask is your permission to publish and distribute it.

UNLESS SPECIFICALLY REQUESTED OTHERWISE, your cassette itself will be a part of your contribution. Except for the most expensive cassettes, it costs the library more in time and money to keep up with and return the cassette than the cassette is worth. Also, the library needs scratch cassettes to help prepare the distribution masters. Radio Shack's Realistic Low Noise - High Frequency cassettes are cheap and work well.

Contributions can be made at any time. The library will send you an acknowledgement of your contribution, and a discount credit which you can use on a future cassette purchase. Or, you can send your contribution with your order.

CONTRIBUTION FORMATS: At the present time, the library is accepting donations in the following formats:

1. PTCo BASIC5 internal compiled format, i.e. SAVE name

2. PTCo ECBASIC internal compiled format, i.e. SAVE name,C

3. PTCo ECBASIC text format, i.e. SAVE name,T

4. Assembler source files as memory images of ALS-8/SOFT1 internal format in block access tape files. Each source line consists of:
   A. The length byte, which gives the length of the entire line, including the length byte and the trailing <CR>,
   B. A four-digit ASCII line number, including leading Ø's,
   C. A space (between the number and the text),
   D. ASCII text source line, and
   E. A carriage-return character, ØDH.

   This file format is written by the SAVE command of the newer versions of SOFT1, and by using SOLOS/CUTER to save a memory image of an ALS-8 file or an older SOFT1 file.

5. Assembler source files and BASIC programs as ASCII text in the PROTEUS STANDARD BYTE ACCESS tape format, described below. The PROTEUS STANDARD BYTE ACCESS format is intended to be the universal bridge between all of the various disk systems. PTCo has given us permission to distribute their PACK and UNPACK programs through the library; these programs allow the conversion between byte access files and block access files. Drivers are being written to allow byte access tapes to be read by the major disk systems and by ALS-8, and these will be published as they become available. We also need output drivers to allow these systems to write byte access files. One such output driver was written by Richard Greenlaw and published in Solus News, V1#3.

6. Executable object code, designed to work under SOLOS/CUTER. Where possible, SOLOS/CUTER should be accessed only through the entry jump table. If necessary to access by calls to internal routines, these should be documented so that users can adapt if necessary.

7. CP/M programs and text files in the block access tape format of the DISKTAPE program written by Richard Greenlaw, and distributed on library cassette C9. So many of our members are now using one of the many varieties of CP/M that we want to offer this as a contribution format. However, most of the programs will be converted to one of the other formats before distribution so that they can be used by all. An exception will be programs which make use of the CP/M disk facilities; others couldn't use them anyway.

8. CP/M files on 8" single density soft-sector diskettes. Again, all applicable programs will be transferred off the disk and distributed on cassette so that all can use them.

CP/M programs and files should have specific application to SOLs or our compatable computers; we don't want to go into competition with the CP/M users group.

DOCUMENTATION: All programs should be fully documented. Where at all possible, the documantation should be on the magnetic media so that the library, and the members, will not have to bear the cost of duplicating hard copy documentation. For BASIC programs, the documentation should be in the form of imbedded REM statements and/or user prompts in PRINT statements. For object programs, the documentation should be in the form of well commented source code with instructions. If you do not wish to make the source public, then separate documentation should be provided or the program should be self prompting. As a last resort, we will accept hard copy documentation.

CORRESPONDENCE: Address all inquiries, orders, contributions, and other correspondence to:

Proteus Cassette Software Library
c/o Lewis Moseley, Jr., Librarian
2576 Glendale Ct., NE
Conyers, GA 30208

If you wish a personal reply, please enclose a self-addressed stamped envelope (SASE). A catalog showing the contents of all of the library cassettes should be available in January, 1980. if you want a copy, send a SASE. The same information has been and/or will be published in PROTEUS NEWS.

HELIOS LIBRARY TO CHANGE HANDS

Because of the increased demands of our subscribers the responsibility for the Helios library is being transfered to Tony Severa. Disk H-4 can be ordered by contacting Tony at 131 Highland Ave, Vacaville, Ca. 95688. Disks H-1 thru H-3 can still be ordered by writing to Stan Sokolow at Proteus until January 1980. At that time all Helios Disk software will be handled by Tony.

To encourage the growth of the library, we have two ways to get the diskettes. First, the preferred way is by sending $10 dollars(US dollars) and an acceptable program for donation to the library, on your Helios diskette. An acceptable program will be defined as any non-copyrighted work of your own creation. Any copyrighted software, such as that taken from a magazine must have the owners written ok for it to be added to our library. If you are donating a program, we ask that you read and complete the copyright statement on the order form at the end of this issue.

Programs are not the only work accepted, tutorial files on any related subject (ie. how to use CP/M, PTDOS, data files, etc) will be acceptable so long as it does not infringe on anyone's copyright. There is a severe need for networking information on subjects like who repairs PT equipment in your area, how to modify PT equipment, troubleshooting techniques, etc.

COPYRIGHTS

All we ask is a statement that you are not donating someone's copyright ed program without appropriate permission if necessary and that you give Proteus license to reproduce the program. (This license is not necessarily exclusive--you can give the program to other users's groups, etc.) You may place your copyright on the program you donate or leave it off, as you see fit. We want to share as much as possible with each other and hope this copyright rule will satisfy you. If not please feel free to contact us with your suggestion and we will be glad to take it into consideration.

GUIDELINES FOR SUBMISSION OF PROGRAMS (FILES) TO HELIOS LIBRARY

1. FILE NAMES: Since the same program can exist in several forms, we have established the following conventions for file names to distinguish between these alternative forms. Users of the library may want to rename the files for their own convenience, but at least the library will be consistent.

A. Names should not contain lower case letters.

B. Each name should end with a suffix that indicates the form of the data contained in the file. For example:

:S Source code, regardless of the language. (The PTDOS file type will indicate the language.)

:O Object code, such as compiled form of BASIC or PASCAL. (The actual object language is indicated bye the file type field. See below about Image files.)

:T Text, not in a programming language. Use this type for data files that are in ASCII, such as a dictionary, a table, a document, etc..

:D Documentation text file which explains how to use the other files having the same prefix name.

:C Contents abstract for addition to the Table of Contents file on the diskette. This is a text file similiar to the :D but very brief. This file will be published in the catalog of library programs.

none No suffix is necessary if the file is an image file, such as a command name, or if it is a device file. You can use the ":O" suffix if you desire.

C. For example, a BASIC program ("PROG") in text form will need the following files: PROG:S, PROG:D, PROG:C. A device driver ("DEVICE") submitted as both source and assembled files of type "D" will need files; DEVICE:S,DEVICE(type D), DEVICE:D, DEVICE:C.

D. The colon in the above examples can be replaced by another punctuation character if you prefer.

2. FILE TYPES: The file types will indicate the language of the progfram. Whenever a PTC file type convention is relevant, it should be used. (See section 3 of the PTDOS manual.)

The following types have been established so far. As with file names, if none of these are appropriate, create your own and we'll add it to the list.

HEX VALUE	SYMBOL	DESCRIPTION
80	00	System files (reserve for PTDOS)
81	01	Numerical data in binary form.
82	02	Numerical data in BCD form.
83	03	Stored FOCAL program.
84	04	Semi-compiled BASIC/5 program.
85	05	Semi-compiled EDBASIC program.
86	06	Source (text) EDBASIC program.
87	07	Serial Access File.
88	08	Random Access Data file.
C1	A	Archive (SAVE) file.
AE	.	Default
A4	$	DO file with command lines.
D4	T	Text file (also BASIC/5 text form)
C0	P	PASCAL source code(text form)
F0	p	PASCAL p-code form (semi-compiled)
00	I00	Image files associated with system

43	IC	Command files.
47	IG	Games (image file).
53	IS	Major sybsystem (compiler, etc).
54	I$	Command for use in DO files.
2A	I.	Default image file.

ALL NEW SUBMISSIONS SHOULD BE SENT TO:
Tony Severa
131 Highland Ave.
Vacaville, Ca. 95688

(Your disks will be returned as soon as possible.)

SOFTWARE WANTED

I am preparing to write a software routine for personal use and for contribution to the Proteus Cassette Software Library. This program will be an interface between the popular CP/M operating system and existing PTCo software. Since I am a firm believer in standing on each others shoulders, rather than on each others toes, I am asking anyone who has already attempted such a program to share his experiences with me.

The program would be called in from disk as a transient program by typing its name followed by the target program name, i.e.

```
A> XEQ ECBASIC.BLK
```

The program would initially do the following five things:

1. Load and execute at the CP/M transient address of 100H,
2. Relocate itself out of the way, perhaps in the C900 range,
3. Find and read in at address 0 the specified disk file,
4. Load registers HL with the address of a surrogate jump table which would replace the usual SOLOS/CUTER jump table, and
5. Begin the program by a jump to address 0.

The surrogate jump table should interface all, or at least some, of the SOLOS/CUTER functions to the CP/M system. The input routines (SINP and AINP) could simply jump to the appropriate points in the SOLOS/CUTER jump table. The output routines could send Pseudoport 0 to the CP/M console and all other to the CP/M lister. The Block Read and Block Write should locate and call in disk files rather than tape files. I suggest that the CP/M file-type .BLK be used for this purpose. A jump to RETRN should reboot CP/M. Eventually, sequential disk I/O routines could be written to simulate the SOLOS/CUTER byte access tape routines.

If you have implemented all or part of a SOLOS/CP/M interface, please write to:
Proteus Cassette Software Library
c/o Lewis Moseley, Jr., Librarian
2576 Glendale Ct. NE
Conyers, Ga. 30208 Thanks.

PROPOSED STANDARD FOR BYTE ACCESS TAPE FILE EXCHANGE

When our User's Society (Proteus, formerly known as Solus) first began, most of us were using the SOL/CUTS block access tape format for most of our programs and data. Because of this common link between all SOL owners (and compatable S-100 systems operating under CUTER), software transfer was simple and easy. However, at this time (late 1979) most of us are using some kind of disk system, and because of this, block access tape is not a universal exchange medium. However, we still have the byte access tape format, and can use it as an exchange medium. Unfortunately, the various disk systems and other operating systems do not necessarily use a compatable internal format. To help overcome this obstacle, the library is proposing the PROTEUS STANDARD BYTE ACCESS FORMAT. The idea is that drivers can be written to allow the various operating systems to write a byte access tape file in a universal format, and that other drivers can be written to read in this format. The format which is being proposed is identical to that used by the user at the keyboard; thus the operating system sees the byte access tape file as a "fast typist".

The WRITE drivers should write the STANDARD BYTE ACCESS tape files in this format:

```
     <line#>  <ASCII text>  <carriage-return>
```

Normally, the WRITE driver would operate as a Custom Output routine under SOLOS/CUTER, and would intercept a listing of the program, and send it to the tape instead of the screen. The WRITE driver should filter out leading spaces (but not leading zeros), line-feeds, nulls, and other control characters.

Similarly, the READ drivers would operate as Custom Input routines, and would feed the tape characters to the operating system when it requests keyboard characters. The READ driver should be able to accept as input the WRITE format specified above without requiring any additional characters. However, the READ driver should be able to tolerate nulls, line-feeds, and leading spaces in case the WRITE driver did not do its job properly. If the operating system requires length bytes, line-feeds, or other special characters, then the driver or the system should provide them. This would usually be done by the system automatically if the READ driver properly simulates the "fast typist". The READ driver should handle end-of-file and tape-read-error conditions, typically by giving a message and switching back to the keyboard for further input.

This format is compatable with the byte access format used by ECBASIC, PTCo EDIT, and the PTCo PACK and UNPACK utilities. Currently, compatable drivers have been written for Micropolis (by Melvin Dalton) and ALS-8 (by Lewis Moseley, Jr.) Drivers for CP/M are being written. The library needs drivers for Northstar and for any other systems which are in use.

The library welcomes comments, suggestions, and especially compatable drivers and software. Address all correspondance to: Proteus Cassette Software Library
c/o Lewis Moseley, Jr., Librarian
2576 Glendale Ct. NE
Conyers, Ga. 30208
If you would like a personal reply, please enclose a SASE.

BERKS READING AREA COMPUTER ENTHUSIASTS MEET

This group has been meeting for quite some time and welcome anyone living in the area to contact them for information on upcoming meetings:

Paul Adams, Cooridinator
Box 210 A RD #4
Reading PA 19606
215-929-2637

Ron Genova, Secretary
601 Friedensburg Rd.
Reading, PA 19606
215-779-5018

Bob Dietzel, Treasurer
4801 Club Drive
Reading, Pa 19606
215-779-7248

ADVANCED COMPUTER PRODUCTS SELLS P.T. EQUIPMENT

Dear Stan,

I was at the Personal Computing Show in New York yesterday and was talking with Mr. Gary Cassity of ADVANCED COMPUTER PRODUCTS about their acquisition of VISTA computers, when I noticed that he had a couple of SOLs on display. It seems that ADVANCED COMPUTER PRODUCTS, PO Box 17329, IRVINE, CA 92713 has purchased the entire Processor Tech enventory of spare parts, work in progress, and finished goods! They are offering this material for sale on a first-come first-served basis. They had SOL-20 without memory for $1295, those beautiful SOL keyboards for $139.95, and 8KRA Assembled boards for $129.95. Not only this, but Mr. Cassity said that they were interested in continuing support on some of the SOL product line. If you specifically ask for it, they will send you a copy of their complete enventory list of Processor Tech parts. I looked over the list he had with him, and he seems to have a good supply of most items. Write for your own copy of the list, and let these people know that we DO WANT support for the SOL!!!

As I mentioned earlier, I also talked with Mr. Cassity about the VISTA drives and controllers. (I just had an extensive article published in the September 1979 issue of KILOBAUD which details how to get a TARBELL or VISTA controller to work with the SOL). I hope in the near future to have the double density controller from VISTA (I have a VISTA single density controller at present, plus a MICROMATION DOUBLER which I use in both single and double density on 8" drives). By the way, I highly recommend the MICROMATION DOUBLER. I have had absolutely no problems with it. They have a BIOS that supports the SOL. I heavily modified mine to add lots of extra features like the ability to change I/O drivers at any time using a control key, but at least the original BIOS was immediately useable if not overly elegant.

Also, for all you DIABLO users, I am enclosing a custom driver for the DIABLO that allows it to run at FULL speed. This is great for those long printouts! It includes echoing of the printed character on the CRT. which is convenient for the operator. I hope this material is useful to a few SOL users at least. When I get the time I will make a listing of my custom CONST, CONIN and CONOUT routines for use with the ELECTRIC PENCIL II DIABLO/SOL version, as well as my routines for use with MICROSOFT EXTENDED DISK BASIC (MBASIC). I have found it advantageous to have a custom BIOS for certain programs due to their peculiarities, and SOL's personality. No sense in having someone duplicate my efforts.

Stan, please enter a subscription for me for this year and send me the bill. I received the back issues I requested, and have given them a place of honor on my shelf. Keep up the good work (and don't dare forget to enter my subscription!)

Sincerely Yours,

Fr. Thomas McGahee

Fr. Thomas McGahee S.D.B.

Fr. Thomas McGahee
Don Bosco Tech
202 Union Ave.
Paterson, N.J. 07502

```
D000           *** CUSTOM DRIVER FOR DIABLO 1610/1620 PRINTER ***
D000           *
D000           *** written by Fr. Thomas McGahee
D000           *** Don Bosco Tech
D000           *** 202 Union Ave.
D000           *** Paterson, N.J., 07502
D000           *
D000           *** SETUP is optional... it automatically makes the
D000           *** DIABLO DRIVER Custom Output Device #3.
D000           *
D000           OPORT  EQU   0C807H  ;SOL Current Output Pseudo Port
D000           UOPORT  EQU  0C802H  ;SOL Output Port Address
D000           *
D000                  ORG   0D000H  ;Adjust this address for your system.
D000           *
D000 E5        SETUP  PUSH  H      ;Allow it to be called by any program
D001 3E 03            MVI   A,3    ;Set SOL parameters so that
D003 32 07 C8         STA   OPORT  ; Output Port #3 is current
D006 21 0E D0         LXI   H,DIABLO  ;Give SOL the address of DIABLO
D009 22 02 C8         SHLD  UOPORT
D00C E1               POP   H      ;Restore registers
D00D C9               RET
D00E           *
D00E           *** DIABLO is the actual DIABLO driver. It prints
D00E           *** from the B register. It also echoes the character
D00E           *** on the CRT. It makes use of the fact that the
D00E           *** DIABLO has a print buffer to allow the DIABLO
D00E           *** to print at MAXIMUM speed. The SOL and DIABLO
D00E           *** should be set to operate at 1200 Baud, though it
D00E           *** will also run at 300 Baud.
D00E           *
D00E           SDROT  EQU   0C04AH  ;SOL Serial Data Out Routine
D00E           SSTAT  EQU   0C042H  ;SOL Serial Status Routine
D00E           CHRCNT EQU   0C816H  ;Systems RAM used to hold Count
D00E           AOUT   EQU   0C01CH  ;SOL Output to device "A"
D00E           *
D00E C5        DIABLO PUSH  B      ;Preserve B,C
D00F AF               XRA   A      ;Echo character in B to CRT
D010 CD 1C C0         CALL  AOUT   ; using SOL AOUT routine.
D013 CD 4A C0         CALL  SDROT  ;Send character to DIABLO
D016 3A 16 C8         LDA   CHRCNT  ;Recover stored character count
D019 3C               INR   A      ;Update the count
D01A 32 16 C8         STA   CHRCNT  ;Store updated count
D01D FE 50            CPI   80     ;80 characters is a full buffer.
D01F DA 36 D0         JC    POPB   ;If less than 80, we are done.
D022 06 03     ETX    MVI   B,3    ;Send the End Of Text character
D024 CD 4A C0         CALL  SDROT  ; to the DIABLO.
D027 CD 42 C0  ACK    CALL  SSTAT  ;Wait for the Diablo
D02A CA 27 D0         JZ    ACK    ; to send a response.
D02D FE 86            CPI   86H    ;Check to see if it is an ACKNOWLEDGE.
D02F C2 27 D0         JNZ   ACK    ;If it is not, keep looking.
D032 AF               XRA   A      ;Set A=0
D033 32 16 C8         STA   CHRCNT  ;and reset Character count.
D036 C1        POPB   POP   B      ;Restore B,C
D037 C9               RET          ;And go home.
```

FOR SALE:

The Marin Computer Center is selling six (6) of its Sol20 Terminal Computers with 24K and Hitachi monitors for $1400 each or $7800 for all. They are in excellent condition - we need computers with color graphics. Call David at (415) 472-2650 or (415) 892-6846 evenings.

UPDATE SERVICE (WRITE TO PROTEUS: 1690 WOODSIDE ROAD, #219'
REDWOOD CITY, CA 94061)

US1..UPDATE FOR EXTENDED CASSETTE BASIC TO REV.A. SEE PROTEUS NEWS V2,#3, P.3. SEND ORIGINAL CASSETTE...............$5

US2..UPDATE FOR PTDOS CISKETTE TO PTDOS 1.5 REV.E. SEE PROTEUS NEWS V2,#3, P.3. SEND ORIGINAL CASSETTE...............$5

US3..UPDATE FOR WORDWIZARD VERSION 4.0 MOD 1. SEE V2, #6, P.4

US4..UPDATE MAILMASTER TO VERSION 3.0 MOD 2...........$10

US5..UPDATE ACCPAC ACCOUNTS PAYABLE TO VERSION 1.0 MOD 2..$10

US6..UPDATE ACCPAC GENERAL LEDGER TO VERSION 1.3 MOD 2........

US7..UPDATE YOUR ACCPAC FINANCIAL REPORTING SYSTEM TO VERSION 1.1 MOD 2..$10

PROPRIETORY SOFTWARE

ORDER FROM PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061

P1..WORDWIZARD WORD PROCESSING SYSTEM FOR THE SOL WITH HELIOS DISK, BY BCG. SEE DESCRIPTION IN PROTEUS NEWS VOL2, #6........

P2..MAILMASTER LIST MANAGEMENT SYSTEM FOR SOL WITH HELIOS DISK, BY BCG. SEE DESCRIPTION IN PROTEUS NEWS VOL 2, #6.............

P3..MIALSORT SUPPLEMENT TO MAILMASTER, BY BCG. SEE DESCRIPTION IN PROTEUS NEWS VOL 2, #6.....................$125 /25 MANUAL

P4..ACCPAC GENERAL LEDGER AND FINANCIAL REPORTING SYSTEM FOR SOL WITH HELIOS DISK, BY BCG. SEE VOL 2, #6...$500 /35 MANUAL

P5..ACCPAC ACCOUNTS/PAYABLE SYSTEM FOR SOL WITH HELIOS DISK BY BCG. SEE VOL 2, #6...........................$500 /35 MANUAL

P6..ACCPAC ACCOUNTS/PAYABLE SYSTEM FOR SOL WITH HELIOS DISK, BY BCG. SEE VOL 2, #6...........................$500 /35 MANUAL

P7..ACCPAC PROGRAMMER'S PACKAGE FOR CREATING CUSTOM ACCPAC PROGRAMS BY BCG. SEE VOL 2, #6................$125 /25 MANUAL

P8..WORDWIZARD, MAILMASTER, AND MAILSORT ALL ORDERED TOGETHER AS A PACKAGE DEAL. THAT IS, P1 THRU P3........$675 /95 MANUAL

P9..P1 THRU P6. ALL ORDERED AS PACKAGE DEAL...$1995 /200 MANUAL

CASSETTE LIBRARY CASSETTES

C1..$18 C2..$18 C3..$18 C4..$18 C5..$18 C6..$18 C7..$18

ORDER FROM "LEWIS MOSELEY, JR. 2576 GLENDALE COURT NE, CONYERS, GA 30208"

HELIOS LIBRARY DISKETTES

H1..$25 H2..$25 H3..$25 H4..$25

ORDER FROM "TONY SEVERA, 131 HIGHLAND AVE, VACAVILLE CA. 95688"

PROTEUS LITERATURE

D1..HELIOS II USERS MANUAL	Loan only
D2..8080 DEBUGGER USERS MANUAL	No more.
D3..EXTENDED DISK BASIC USERS MANUAL	Loan only
D4..VDM-1 DISPLAY MODULE USERS MANUAL	No more.
D5..CASSETTE FOCAL USERS MANUAL	No more.
D6..ADVANCED 8080 ASSEMBLER USERS MANUAL	No more.
D7..BASIC/5 USERS MANUAL	Loan only
D8..SOL TERMINAL COMPUTER USERS MANUAL	Loan only
D9..SOFTWARE #1 MANUAL	No more.
D10.5K BASIC SOFTWARE #2 SOURCE LISTING	No more.
D11.SOLOS CUTER USERS MANUAL	$5
D12.CUTS COMPUTER USERS MANUAL	$5
D13.GPM AND GPM-SOL ASSY AND TEST INSTR.	$5
D14.SUBSYSTEM B USERS MANUAL	$5
D15.CASSETTE PILOT UPDATE 731060	$.55
D16.CASSETTE PILOT UPDATE 731069	$1
D17.SOFTWARE #1 UPDATE 731070	$.85
D18.EXTENDED DISK BASIC UPDATE 731062	$.70
D19.VDM-1 UPDATE 731063	$.70
D20.EXTENDED CASSETE BASIC UPDATE 731064	$1.58
D21.EXT. DISK BASIC UPDATE 731065	$.55
D22.8080 CHESS UPDATE 731043	$.85
D23.PTDOS UPDATE 731072	$2.03
D24.PTDOS AND WORDWIZARD UPDATE 731074	$1
D25.WORDWIZARD UPDATE 731075	$1.58
D26.16KRA USERS MANUAL	$5
D27.32KRA USERS MANUAL	$.55
D28.32KRA-1 USERS MANUAL	$5
D30.ACCESS, MARCH 1977	$1
D31.THE SMALL COMPUTER CATALOG	$1
D32.8080 FOCAL USERS MANUAL	No more.
D33.SOFTWARE #2 USERS MANUAL	No more.
D34.ALS8 USERS MANUAL	$7
D36.ALS8 MANUAL CHANGE #2	$.85
D37.EXT. DISK BASIC REF. CARD	$.70
D38.CUTS, ASSY AND TEST INST.	No more.
D39.CUTER SOURCE LISTING	No more.
D40.FORTRAN UPDATE 731077	$4.41
D41.FORTRAN UPDATE 731040A	$.70
D42.SOL MANUAL ADDENDUM 1 & 3	No more.
D43.SOL COMPUTER SYSTEMS MANUAL	Loan only.

D44.ENG. CHANGE #7	$1.30
D45.SOL MANUAL CHANGE #9	$1
D46.SOL MANUAL CHANGE #10	$.55
D47.SOL MANUAL CHANGE #11	$1.30
D48.SOL MANUAL CHANGE #13	$.55
D49.SOL MANUAL CHANGE #14	$.55
D50.SOL MANUAL CHANGE #16	$.70
D51.SOL UPDATE 731049	$1.15
D52.ASSY PROC CHANGE #HSC	$.55
D53.8KRA ASSY AND TEST INST.	Loan only.
D54.ASSY CHANGE #6-2 REV C	$.70
D55.SOL HYTYPE INTERFACE UPDATE 731076	$1.55
D56.NKRA INSTALLATION GUIDE	$3
D57.16KRA &32KRA UPDATE 731066	$.70
D58.64KRA USERS MANUAL	$1
D59.NKRA UPDATE 731047	$1.73
D60.16KRA DIAGNOSTIC TEST	$3
D61.16KRA CHANGE #2	$.70
D62.16KRA & 32KRA UPDATE 731041	$.70
D63.16KRA & 32KRA UPDATE 731042	$.55
D64.HELIOS UPDATE 731067	$.70
D65.PTDOS UPDATE 731073	$1.15
D66.PARA SOL DEBUGGERS USERS MANUAL	$15
D67.HELIOS UPDATE 711048	$.85
D68.HELIOS UPDATE 731071	$.85
D69.FIELD RETROFIT NOTICE 4/1/79 HELIOS	$1
D70.FIELD RETROFIT NOTICE 4/19/79 HELIOS	$.70
D71.PERSCI PCB ASSY LOGIC SCHEMATIC	$5
D72.HELIOS II DISK SYSTEM MANUAL	$25
D73.HELIOS II MANUAL CHANGE #3	$.85
D74.HELIOS II MANUAL CHANGE #4	$.85
D75.REV. A OLD LOOSE LEAF HELIOS MANUAL	$1.73
D76.WORDWIZARD UPDATE #3	$2.00
D77.ALS8 NEWSLETTER V1#1	$7
D78.CHAPTER VIII FOR SOL SYSTEM MANUAL	$2.03
D79.DIABLO SERIES 2300 MAINT. MANUAL	$2.03
D80.DIABLO SYSTEM MAINT. AND PRICING MANUAL	$7

PLEASE ADD $1.00 PER COMPLETE ORDER TO COVER HANDLING AND POSTAGE. MAKE CHECKS PAYABLE TO "TONY SEVERA" AND MAIL TO 131 HIGHLAND AVE. VACAVILLE CA 95688 (ON LOANS CALL 707-446-0417 FOR INSTRUCTIONS) ANYONE WANTING TO DONATE ANY MISSING ITEMS FROM THE ABOVE LIST OR ANY OTHER PROCESSOR TECH MANUAL OR UPDATE PLEASE CONTACT TONY SEVERA.

(See Vol2 #4 for more detail on these items)

WORLD GENERATOR by Tony Severa

```
  100 REMPROGRAM TO MAKE RANDOM WORLDS.
THIS PROGRAM WRITTEN FOR THOSE OF YOU WHO ARE INTO D & D OR
FANTASY ROLE PLAYING GAMES. THIS PROGRAM, WHEN RUN WILL DESIGN
FANTASY WORLDS FOR GAMES LIKE TRAVELER, GAMMAWORLD OR D&D.
ENJOY!
  200 REM
  300 REMMAP LOCAL SUBSECTOR
  400 REMIS WORLD PRESENT
  500 REMSPACE LANES: JUMP ROUTES
  600 REMSTAR PORT TYPES
  700 LET S=INT(RND(0)*6)+1
  800 IF S<2 THEN  GOSUB 2200
  900 IF S=2 THEN  GOSUB 2700
 1000 IF S=3 THEN  GOSUB 3200
 1100 IF S=4 THEN  GOSUB 3500
 1200 IF S=5 THEN  GOSUB 3800
 1300 IF S=6 THEN  GOSUB 4200
 1400 REMSPECIFIC WORLDS.
 1500 REMNOTE STARPORT TYPE
 1600 REMGENERATE PLANETARY SIZE
 1601 PRINT "PLANETARY SIZE"
 1605 LET P=13*INT(RND(0)+1)
 1610 LET P=INT(RND(0)*13)+1
 1615 IF P=2 THEN  PRINT "1000 MILES DIAMETER."
 1620 IF P=3 THEN  PRINT "2000 MILES DIAMETER."
 1625 IF P=4 THEN  PRINT "3000 MILES DIAMETER."
 1630 IF P=5 THEN  PRINT "4000 MILES DIAMETER."
 1635 IF P=6 THEN  PRINT "5000 MILES DIAMETER."
 1640 IF P=7 THEN  PRINT "6000 MILES DIAMETER."
 1645 IF P=8 THEN  PRINT "7000 MILES DIAMETER."
 1650 IF P=9 THEN  PRINT "8000 MILES DIAMETER."
 1655 IF P=10 THEN  PRINT "9000 MILES DIAMETER."
 1660 IF P=11 THEN  PRINT "10000 MILES DIAMETER."
 1665 IF P=12 THEN  PRINT "11000 MILES DIAMETER."
 1670 IF P>12 THEN  PRINT "12000 MILES DIAMETER."
 1700 REMGENERATE PLANETARY ATMOSPHERE
 1705 LET P=INT(RND(0)*13)+1
 1706 PRINT "            PLANETARY ATMOSPHERE"
 1710 ON P GOTO 1720,1725,1730,1735,1740,1745,1750,1755,1760,176
5,1770,1775,1780
 1720 PRINT " NO ATMOSPHERE": GOTO 1800
 1725 PRINT " TRACE OF ATMOSPHERE": GOTO 1800
 1730 PRINT " VERY THIN, TAINTED.": GOTO 1800
 1735 PRINT " THIN, TAINTED.": GOTO 1800
 1740 PRINT " VERY THIN.": GOTO 1800
 1745 PRINT " STANDARD ": GOTO 1800
 1750 PRINT " STANDARD ": GOTO 1800
 1755 PRINT " STANDARD, TAINTED.": GOTO 1800
 1760 PRINT " DENSE.": GOTO 1800
 1765 PRINT " DENSE, TAINTED.": GOTO 1800
 1770 PRINT " EXOTIC": GOTO 1800
 1775 PRINT " CORROSIVE.": GOTO 1800
 1780 PRINT " INSIDIOUS.": GOTO 1800
 1800 REMGENERATE POPULATION
 1805 LET P=INT(RND(0)*11)+1
 1806 PRINT "                  POPULATION"
 1810 ON P GOTO 1815,1820,1825,1830,1835,1840,1845,1850,1855,186
0,1865
 1815 PRINT " 0 INHABITANTS.": GOTO 1900
 1820 PRINT " 10 INHABITANTS. ": GOTO 1900
 1825 PRINT " 100 INHABITANTS. ": GOTO 1900
 1830 PRINT " 1000 INHABITANTS. ": GOTO 1900
 1835 PRINT " 10,000 INHABITANTS. ": GOTO 1900
 1840 PRINT " 100,000 INHABITANTS. ": GOTO 1900
 1845 PRINT " 1,000,000 INHABITANTS. ": GOTO 1900
 1850 PRINT " 10,000,000 INHABITANTS. ": GOTO 1900
 1855 PRINT " 100,000,000 INHABITANTS. ": GOTO 1900
 1860 PRINT " 1,000,000,000 INHABITANTS. ": GOTO 1900
 1865 PRINT " 10,000,000,000 INHABITANTS. ": GOTO 1900
 1900 REMGENERATE GOVERNMENT
 1905 LET G=INT(RND(0)*14)+1
 1906 PRINT "                GOVERNMENTAL TYPE"
 1910 ON G GOTO 1915,1920,1925,1930,1935,1940,1945,1950,1955,196
0,1965,1970,1975,1980
 1915 PRINT "NO GOVERNMENT STRUCTURE. FAMILY BONDS.": GOTO 2000
 1920 PRINT " COMPANY/CORPORATION. RULILNG FUNCTIONS ARE ASSUMED
 BY A"
 1921 PRINT " COMPANY MANAGERIAL ELITE, AND MOST CITIZENS ARE EM
PLOYEES"
 1922 PRINT "OR DEPENDANTS.": GOTO 2000
 1925 PRINT " PARTICIPATING DEMOCRACY. RULING FUNCTION DECISIONS
 ARE"
 1926 PRINT " REACHED BY THE ADVICE AND CONSENT OF CITIZENS DIRE
CTLY."
```

```
 1927 GOTO 2000
 1930 PRINT " SELF-PERPETUATING OBLGARCHY. RULING FUNCTIONS ARE
PERFORMED"
 1931 PRINT " BY A RESTRICTED MINORITY, WITH LITTLE OR NO  INPUT
 FROM CITIZENS."
 1932 GOTO 2000
 1935 PRINT " REPRESENTATIVE DEMOCRACY. RULING FUNCTIONS AR EPER
FORMED "
 1936 PRINT " BY ELECTED REPRESENTATIVES.": GOTO 2000
 1940 PRINT " FEUDAL TECHNOCRACY. RULING FUNCTIONS ARE PERFORMED
 BY"
 1941 PRINT " SPECIFIC INDIVIDUALS FOR PERSONS WHO AGREE TO BE R
ULED BY"
 1942 PRINT " THEM. RELATIONSHIPS ARE BASED ON THE PERFORMANCE O
F TECHNICAL"
 1943 PRINT " ACTIVITIES WHICH ARE MUTUALLY BENEFICIAL.": GOTO 2
000
 1945 PRINT " CAPTIVE GOVERNMENT. RULING FUNCITONS ARE PERFORMED
 BY AN"
 1946 PRINT " IMPOSED LEADERSHIP ANSWERABLE TO OUTSIDE GROUP. A
COLONY OR"
 1947 PRINT " CONQUERED AREA.": GOTO 2000
 1950 PRINT " BALKANIZATION. NO CENTRAL RULING AUTHORITY EXISTS;
 RIVAL"
 1951 PRINT " GOVERNMENTS COMPETE FOR CONTROL. ": GOTO 2000
 1955 PRINT " CIVIL SERVICE BUREAUCRACY. RULING FUNCTIONS ARE PE
RFORMED"
 1956 PRINT " BY GOVERNMENT AGENCIES EMPLOYING INDIVIDUALS SELEC
TED FOR "
 1957 PRINT " THEIR EXPERTISE."
 1958 GOTO 2000
 1960 PRINT " IMPERSONAL BUREAUCRACY. RULING FUNCITONS ARE PERFO
RMED BY"
 1961 PRINT " AGENCIES WHICH HAVE BECOME INSULATED FROM THE GOVE
RNED "
 1962 PRINT " CITIZENS.": GOTO 2000
 1965 PRINT " CHARISMATIC DICTATOR. RULING FUNCTIONS ARE PERFORM
ED BY "
 1966 PRINT " AGENCIES DIRECTED BY A SINGLE LEADER WHO ENJOYS TH
E OVER-"
 1967 PRINT " WHELMING CONFIDENCE OF THE CITIZENS.": GOTO 2000
 1970 PRINT " NON-CHARISMATIC LEADER. A PREVIOUS CHARISMATIC DIC
TATOR HAS"
 1971 PRINT " BEEN REPLACED BY A LEADER THROUGH NORMAL CHANNELS.
": GOTO 2000
 1975 PRINT " CHARASMATIC OLIGARCHY. RULING FUNCTIONS ARE PERFOR
MED BY A"
 1976 PRINT " SELECT GROUP OF MEMBERS OF AN ORGANIZATION OR CLAS
S WHICH "
 1977 PRINT " ENJOYS THE OVERWHELMING CONFIDENCE OF THE CITIZENR
Y.": GOTO 2000
 1980 PRINT " RELIGEOUS DICTATORSHIP. RELIGIOUTS ORGANIZATION WI
THOUT "
 1981 PRINT " REGARD TO THE SPECIFIC INDIVIDUAL NEEDS. ": GOTO 2
000
 2000 REMGENERATE LAW LEVEL
 2005 LET L=INT(RND(0)*10)+1
 2006 PRINT "                        LAW LEVELS"
 2010 ON L GOTO 2015,2020,2025,2030,2035,2040,2045,2050,2055,206
0
 2015 PRINT " NO LAWS AFFECTING WEAPONS POSSESSION OR WEAPONS OW
NERSHIP.": GOTO 2100
 2020 PRINT " CERTAIN WEAPONS ARE PROHIBITED, INCLUDING 1.BODY P
ISTOLS AND"
 2021 PRINT " 2. EXPLOSIVE WEAPONS AND 3. POISEN GAS.": GOTO 210
0
 2025 PRINT " PORTABLE ENERGY WEAPONS, SUCH AS LASER RIFLES OR C
ARBINES"
 2026 PRINT " ARE PROHIBITED. SHIP'S GUNNERY IS NOT AFFECTED.":
GOTO 2100
 2030 PRINT " WEAPONS OF A STRICT MILITARY NATURE ARE PROHIBITED
.": GOTO 2100
```

```
 2035 PRINT " LIGHT ASSAULT WEAPONS ARE PROHIBITED.": GOTO 2100
 2040 PRINT " PERSONAL CONCEALABLE FIREARMS ARE PROHISITED.": GO
TO 2100
 2045 PRINT " MOST FIREARMS ARE PROHIBITED. THE CARRYING OF ANY
TYPE OF"
 2046 PRINT " WEAPON OPENLY IS DISCOURAGED.": GOTO 2100
 2050 PRINT " SHOTGUNS ARE PROHIBITED.": GOTO 2100
 2055 PRINT " LONG BLADED WEAPONS ARE STRICTLY CONTROLLED. OPEN
POSSESSION"
 2056 PRINT " IN PUBLIC IS PROHIBITED. OWNERSHIP IS, HOWEVER, NO
T RESTRICTED."
 2057 GOTO 2100
 2060 PRINT " POSSESSION OF ANY WEAPON OUTSIDE OF ONE'S HOME IS
PROHIBITED.": GOTO 2100
 2100 REMGENERATE TECHNOLOGICAL INDEX
 2105 PRINT "                        TECHNOLOGICAL INDEX"
 2106 LET T=INT(RND(0)*12)+1
 2110 ON T GOTO 2115,2120,2125,2130,2135,2140,2145,2150,2155,216
0,2165,2170,2175
 2115 PRINT " CLUBS.CUDGEL..SPEARS..COMMUNICATIONS BY RUNNERS."
 2116 PRINT " CANOES...CARTS": GOTO 2199
 2120 PRINT " DAGGERS..PIKE..SWORD..JACK..CATAPU LT..ABACUS..HEL
IOGRAPH"
 2121 PRINT " GALLEYS..WAGONS": GOTO 2199
 2125 PRINT " HALBERDS..BROADSWORDS..CANNONS..:": GOTO 2199
 2130 PRINT " FOIL, CUTLASS, BLADE, BAYONET..SAILING SHIPS..HOT
AIR BALOONS": GOTO 2199
 2135 PRINT " REVOLVER..SHOTGUN..CLOTH ARMOR..ARTILLERY..TELEPHO
NES..STEAM"
 2136 PRINT " SHIPS..TRAINS..DIRIGIBLES..COAL POWER.": GOTO 2199
 2140 PRINT " CARBINE..RIFLE..PISTOL..MORTARS..COMPUTER MODEL 1.
.RADIO"
 2141 PRINT " GROUND CARS..FIXED WING AIRCRAFT..OIL POWER.": GOT
0 2199
 2145 PRINT " AUTO RIFLE..MISSILES..ROCKET LAUNCHERS..MODEL 1 CO
MPUTER.."
 2146 PRINT " TELEVISION..SUBMERSIBLES..ROTARY WING AIRCRAFT..FI
SSION.": GOTO 2199
 2150 PRINT " BODY PISTOL..MESH..PULSE LASER..MODEL 2 COMPUTERS.
.AIR/RAFTS"
 2151 GOTO 2199
 2155 PRINT " LASER CARBINE..AUTO-CANNON..MODEL 2 COMPUTERS..NON
-STARSHIPS.": GOTO 2199
 2160 PRINT " LASER RIFLES..ABLAT..BEAM LASER..MODEL 3 COMPUTERS
..STARSHIPS": GOTO 2199
 2165 PRINT " REFLEC ARMOR..MODEL 4 COMPUTERS..STARSHIPS.": GOTO
 2199
 2170 PRINT " LASER RIFLES..BEAM LASERS..MODEL 4 COMPUTERS..STAR
SHIPS.": GOTO 2199
 2175 PRINT " REFELC ARMOR..MODEL 5 COMPUTERS..STARSHIPS.": GOTO
 2199
 2199 PRINT : PRINT : PRINT : PRINT : GOTO 100
 2200 REM**STARPORT SUBROUTINE
 2300 PRINT "EXCELLENT QUALITY INSTALLATION. REFINED FUEL AVAILA
BLE."
 2400 PRINT "ANNUAL MAINTENACE OVERHALL AVAILABLE. SHIPYARD CAPA
BLE OF"
 2500 PRINT "BOTH STARSHIP AND NON-STARSHIP CONSTRUCTION PRESENT
. "
 2600 RETURN
 2700 REM**STARPORT SUBROUTINE
 2800 PRINT "GOOD QUALITY INSTALLATION. REFINED FUEL AVAILABLE.
"
 2900 PRINT "ANNUAL OVERHALL AVAILABLE. SHIPYARD CAPABLE OF CONS
TRUCTING"
 3000 PRINT "NON-STARSHIPS PRESENT."
 3100 RETURN
 3200 REM**STARPORTS SUBROUTINE
 3300 PRINT "ROUTINE QUALITY INSTALLATION. ONLY UNREFINED FUEL A
VAILAB"
 3400 RETURN
 3500 PRINT "POOR QUALITY INSTALLATION. ONLY UNREFINED FUEL AVAI
LABLE."
 3600 PRINT "NO REPAIR OR SHIPYARD FACILITIES PRESENT."
 3700 RETURN
 3800 REM*STARPORT SUBROUTINES
 3900 PRINT "FRONTIER INSTALLATION. ESSENTIALLY A BARE SPOT OF B
EDROCK" /
 4000 PRINT "WITH NO FUEL, FACILITIES, OR BASES PRESENT."
 4100 RETURN
 4200 REM**STARPORT SUBROUTINE
 4300 PRINT "NO STARPORT. NO PROVISION IS MADE FOR ANY STARSHIP
LANDINGS."
 4400 RETURN
```

P.T. LUNAR MODIFIED FOR REAL TIME OPERATION

```
    1 REM..MODIFIED BY TONY SEVERA SO THAT IT OPERATES AS A REAL
    2 REM..TIME GAME.
   10 DIM R$(1)
   20 PRINT "\K": CURSOR 7,0
   30 SET DS=8: PRINT TAB(17);"<<= LUNAR LANDER =>> Real Time"
   31 SET DS=8: PRINT TAB(17);"MODIFIED TO OPERATE IN REAL TIME"
   40 PRINT : SET DS=15: INPUT (1,0)"INSTRUCTIONS?  ",R$
   50 IF R$="N" THEN 270
   60 IF R$<>"Y" THEN 710
   70 SET CM=0
   80 SET DS=3: PRINT "\M\JMISSION CONTROL CALLING LUNAR MODULE!
!!"
   90 PAUSE 20
  100 PRINT "MISSION CONTROL CALLING LUNAR MODULE!!!  DO YOU COP
Y???"
  110 PAUSE 20
  120 PRINT "\J\MCOMPUTER GUIDANCE FAILURE!!! MANUAL OVERRIDE NE
CESSARY!!!"
  130 PAUSE 20
  140 PRINT : PRINT "MANUAL OVERRIDE WILL ALLOW YOU TO RESET BUR
N RATE EVERY 10"
  150 PRINT "SECONDS TO ANY RATE BETWEEN 8 AND 200 LBS/SEC."
  160 PRINT "A BURN RATE OF 0 WILL RESULT IN A FREE FALL FOR 10
SECONDS."
  170 PRINT "AS OF NOW YOUR STATUS IS...."
  190 PRINT TAB(20);: SET CP=1: PRINT " CAPSULE WEIGHT IS 33000
LBS. "
  195 SET CP=0
  200 PRINT TAB(20);: SET CP=1: PRINT " AMOUNT OF FUEL IS 26500
LBS. "
  210 PRINT : SET CP=0
  220 PRINT TAB(10);"ESTIMATED FREE FALL TIME TO IMPACT IS 120 S
ECS."
  230 PRINT : PRINT "GOOD LUCK, YOU'LL NEED IT!!\J";
  240 PRINT "MISSION CONTROL OVER AND OUT."
  250 SET CM=1
  260 PAUSE 20
  270 SET DS=0: PRINT
  275 LET K1$="0"
  280 PRINT "TIME           ALTITUDE        VELOCITY       FUEL
   BURN RATE"
  290 PRINT "SECS    MILES     FEET         MPH            LBS
   LBS/SEC"
  300 LET L=0: LET A=120: LET V=1: LET M=43000: LET N=16500
  310 LET G=.001: LET Z=1.8
  320 PRINT %3I;INT(L);TAB(8);%4I;INT(A);TAB(18);INT(5280*(A-INT
(A)));
  330 PRINT %8F2;TAB(27);3600*V;TAB(41);M-N;TAB(54);
  340 INPUT ,(1,50)"",K$
  345 IF K$="" THEN  LET K$=K1$: PRINT K$;: GOTO 360
  346 INPUT ,(3,15)"",K1$
  347 LET K$=K$+K1$
  350 IF K$="STOP" THEN 710
  360 ERRSET 430
  370 LET K=VAL(K$)
  375 LET K1$=K$
  380 ERRCLR
  385 PRINT
  390 LET T=10
  400 IF K=0 THEN 440
  410 IF K<8 THEN 430
  420 IF K<=200 THEN 440
  430 PRINT "ENTER A BURN RATE OF 0, 8 THRU 200 OR 'STOP' ONLY."
;TAB(54);: GOTO 340
  440 IF (M-N)<.001 THEN 540
  450 IF T<.001 THEN 320
  460 LET S=T
  470 IF M>=N+S*K THEN 490
  480 LET S=(M-N)/K
  490 GOSUB 750
  500 IF I<0 THEN 820
  510 IF V<=0 THEN 530
  520 IF J<0 THEN 860
  530 GOSUB 810: GOTO 440
  540 PRINT "FUEL EXHAUSTED AT";L;" SECONDS."
  550 LET S=(-V+SQR(V*V+2*A*G))/G
  560 LET V=V+G*S: LET L=L+S
  570 PRINT
  571 PRINT "ON THE MOON AT";TAB(20);L;" SECONDS."
  580 LET W=3600*V
  590 PRINT "IMPACT VELOCITY OF";TAB(19);W;" MPH."
  600 LET D2=M-N: IF D2<0 THEN  LET D2=0
  610 PRINT "FUEL REMAINING IS";TAB(20);D2;" LBS."
  615 PRINT
  620 IF W>=60 THEN 670
  630 IF W>=25 THEN  PRINT "CRAFT DAMAGED!!! HOPE YOU CAN MAKE I
T HOME.": GOTO 700
  640 IF W>=10 THEN  PRINT "CONGRATULATIONS ON A POOR LANDING!":
 GOTO 700
  650 IF W>=1 THEN  PRINT "COULD HAVE BEEN A BETTER LANDING, BUT
 IT WAS OK.": GOTO 700
  660 PRINT "A PERFECT LANDING!!!": GOTO 700
  670 PRINT "SORRY, BUT THERE WERE NO SURVIVORS."
  680 PRINT "IN FACT YOU BLASTED A NEW LUNAR CRATER";W*.277777;"
 FEET DEEP."
  690 PRINT "WELL AT LEAST THEY WILL NAME IT IN YOUR MEMORY."
  700 PRINT : PRINT : PRINT
  710 INPUT (1,0)"WOULD YOU LIKE TO TEMPT FATE ONCE MORE?  ",R$
  720 IF R$="Y" THEN 270
  730 IF R$<>"N" THEN 710
  740 PRINT : PRINT : PRINT "        MISSION CONTROL OUT.": END
  750 LET Q=S*K/M
  760 IF Q>=.000001 THEN 780
  770 LET Q=0
  780 LET J=V+G*S-Z*Q*(1+Q*(.5+Q*(1/3+Q*(.25+Q/5))))
  790 LET I=A-G*S*S/2-V*S+Z*S*Q*(.5+Q*(1/6+Q*(1/12+Q/20)))
  800 RETURN
  810 LET L=L+S: LET T=T-S: LET M=M-S*K: LET A=I: LET V=J: RETUR
N
  820 IF S<.005 THEN 570
  830 LET D=V+SQR(V*V+2*A*(G-Z*K/M))
  840 LET S=2*A/D
  850 GOSUB 750: GOSUB 810: GOTO 820
  860 LET W=(1-M*G/(Z*K))/2
  870 LET S=M*V/(Z*K*(W+SQR(W*W+V/Z)))+.05
  880 GOSUB 810: IF I<=0 THEN 820
  890 GOSUB 810
  900 IF J>0 THEN 440
  910 IF V>0 THEN 860
  920 GOTO 440
```

TABLE OF CONTENTS

Continued from Page 1.

High School their status as to what type of micro they may have and if they have a Sol, turn them on to us and/or turn us on to them. We'll send them information on what we are doing and maybe they can share some of their student written software with the rest of PROTEUS members.

When I picked up all the literature destined for print from Stan I noticed that there was a need for more information. You don't have to be a good writer to send us information you think' someone else could use. You'd be surprised how some little article can help someone else. Please send in any ideas, articles, tutorials, programs and reactions to us for inclusion in the newsletter. Don't worry, if you don't want your name attached to it, we'll be glad to omit it. The main thing is that you are sharing your knowledge with someone else.

Notify me of local repair stations for P.T. equipment and try and include price lists. If you want to sell or buy equipment please notify us for inclusion in your classifieds column. Its a free service to all.

I see a need for more articles oriented towards the beginner or novice; especially around the use of PTDOS and its many capabilities. There a lot of command I still do not know or use because I just don't have the time or energy to try and figure out. The latest one I have had to study was the use of an archive file (Get and Save commands). There have been changes to PTDOS and those changes need to be explained by the people who use them the most, you. I would love to see articles on the use of the Assembler, Debugger, Focal and EDT3 with simple(?) examples of how they work. I would also like to see more in the way of examples of how to use Wordwizard commands. Although the manual is well written, it still lacks in its ability to teach beginners.

PROTEUS/NEWS

Publisher: PROTEUS, 1690 Woodside Road, Suite 219, Redwood City, California 94061, U.S.A.

Editor: Tony Severa, 131 Highland Ave, Vacaville, California 95688, U.S.A.

Subscriptions: $18 per calendar year in U.S.A., $18 in Canada, $24 elsewhere (U.S. dollars only). All subscriptions expire at the end of the calendar year (Dec. 31). Mid-year subscribers will receive back issues for current year. Back issues for prior years are sold in annual sets, while available.

Instructions to contributors: Letters and articles may be submitted in camera-ready form or on Sol/Cuts cassettes or Helios (PTDOS) diskettes. Camera-ready copy should be single-spaced, in a single column of 6 1/2 inch width, and with clean, dark type. Corrections can be made invisibly with opaque correction fluid ("liquid paper"). Please use a new ribbon. Machine-readable articles should be compatible with Solos, Cuter, PTDOS input routines. Media will be returned only if requested.

PROTEUS
1690 WOODSIDE ROAD, SUITE 219
REDWOOD CITY, CALIFORNIA 94061
U.S.A.

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

18104

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

April /May 1980 PUBLISHED BIMONTHLY BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA SINGLE ISSUE #3

EDITORIAL

Well, it's time for issue #2 and I've barely had a chance to recover from issue #1! This has been one busy year so far! The 5th West Coast Computer Faire is over and there is lot of information to share with you. I went to the faire wearing several hats. I was selling my software for APPLE ORCHARD; I was keeping PROTEUS's presence there by sharing my booth with PROTEUS; and I was also there to attempt to set up an INTERNATIONAL NORTH STAR USER'S ASSOCIATION. On top of all that I was to coordinate a three hour discussion oriented to NOVICES on computers. Needless to say I lost my voice late Saturday afternoon.

The faire was a success for all and since then my phone hasn't quit ringing. Many of you who call could help me by sending me a letter if your reason for calling isn't urgent. I have been so busy answering calls that I have a hard time getting my normal work done. If you're call is of an urgent manner, by all means, call me! I took three days off last week and my answer box received over 3Ø phone calls in that time period. As you can see it can get quite hectic. Follow-ups on phone calls also have a way of getting lost since I have only a small piece of paper to deal with. Letters get logged and followed, thus receiving better follow-up.

I have found that many of you consider yourselves NOVICES and I want you to know that there are several articles in the working that will help you learn what many of the ins-and-outs of the Sol/Helios/NorthStar systems and also some about the basics of computers. We start this issue with a nice article on using the ASSM (Assembler) command in PTDOS. Those of you who can help write BEGINNER oriented articles, please get them in. We can use all the help you can give us, such as: How do you use the ARCHIVE capability using SAVE and GET?

For those hackers out there we will continue to print articles on I/O routines, modifications, ect. If you have learned something new about your system, write it down on a sheet and send it in. If we need to rewrite it, we will. Also, we may have lost some of your earlier submissions during the changeover of EDITORS so, if it has been over 4 issues since you have sent in your article/letter please send it in again. Last issue I forgot to print an updated version of the Cassette Library Instructions. This was my error and I apologize for any inconvenience it may have caused. Those of you who have not used the cassette library should reconsider it. They have a nice assortment of software and it can alwayse be put on disk. We hope to print a catalog in our next newsletter.

Thank you.

Tony Severa

An Introduction to ASSM--the PTDOS Assembler

by Jay Parsons

The first question about the PTDOS assembler, or the whole subject of assembly language, is doubtless "Why bother?" There are arcane reasons such as speed and memory utilization, but the main one is that some programs cannot be implemented efficiently any other way.

As an example, let's consider the problem of defeating the attribute protection of PTDOS files--so that we might, for example, remove information protection from a semi-compiled BASIC program. Joe Maguire gives the answer in Vol. 2, #5 of PROTEUS / News: put a binary ØØ in memory location A89Ø (the location is A8B1 for PTDOS 1.4; from now on PTDOS 1.5 will be assumed) before using REATR. This can be accomplished by exiting to SOLOS and using the ENter command, but that requires some time, accurate typing and, worst of all, remembering the address to patch, the byte to patch it with, and how to reenter PTDOS when finished. A one-line BASIC program (e. g.: 1Ø POKE 43152,Ø:BYE) would suffice, but it is slow and rather silly to load and run an umpteen-K BASIC interpreter for such a purpose.

By writing and assembling a little assembly-language program, we can have a program that makes the necessary patch quickly and that can be run from PTDOS merely by typing its name (let's say ZAP) after the PTDOS asterisk prompt, or even by placing the name in a macro to be executed by DO or SETIN.

An assembly-language program involves three distinct steps: writing the source code, assembling it and running the resulting assembled or "object" code. In practice there's always a fourth step, debugging the program, but that's another story.

Step 1 -- The Source Code

We've decided to create a program that will run as "ZAP"; to do so we must first write the necessary assembly-language instructions, or source code, to a PTDOS file. The source code is essentially text. It can be written using any text-editing program available, such as Word Wizard or Electric Pencil, but because the resulting file must contain "lines" separated by carriage returns I find it easiest to use EDIT, which automatically creates a file with the correct format. We need a name other than "ZAP" for the source file; let's use "ZAP:S". I'm lazy, so I would commence step 1 by typing "EDIT ZAP:S <cr>"; meticulous types may prefer to CREATE ZAP:S first and to give it an appropriate type and block size, such as type T and 1ØØH.

Cont. Pg. 2

Cont. from Pg. 1.

Moving right along, we type another <cr> in response to EDIT's prompt and see a screen full of number signs. Panic sets in. Easy to say "write the source code", but how? Here I must cop out; this is not a programming course. Assembly language programs are written using mnemonics assigned by the CPU manufacturer (in this case Intel Corporation, maker of the 8Ø8ØA) for each operation the CPU is to perform, and you'll simply have to learn them and what they do. Appendix 4 to the ASSM manual (bound in the PTDOS 1.5 manual) gives the two classic references, Intel's excellent manual describing each mnemonic and giving some programming examples and Lance Leventhal's text; I found How to Program Microcomputers, by William Barden, Jr., Howard W. Sams & Co., Inc., Indianapolis, 1977, to be a useful introduction.

Mnemonics aside, creating the source file requires familiarity with the requirements of the assembler, ASSM, as well as with EDIT or whatever program is being used to edit the text. ASSM imposes format requirements on each line (discussed in Section 2 of the ASSM manual) as well as on the file as a whole; it also requires that all labels used be properly defined and that each word in the "operation" field be either a mnemonic or one of the "pseudo-ops" described in Section 3 of the ASSM manual, such as ORG, XEQ, EQU or DB.

One of the functions of the assembler is to keep track of where in memory each instruction will be loaded; it does the counting, but we have to tell it where to load the first byte. We do this by placing an ORG statement in the source file.

Assembly programmers do a lot of thumbing through the PTDOS manual; suffice it to say that Section 6.2 of the manual mentions the desirability of running short system command programs in the command execution buffer part of memory beginning at CXBUF, so they won't interfere with anything else in memory. Anything that runs by typing its name to PTDOS is a system command for this purpose; since only one will be running at a time they may be loaded into the same memory area in turn.

Our ZAP program will be short, so we bravely, and at long last, start our source file with the statement " ORG CXBUF." At least one space before the "ORG" is required so the assembler will know it is an operation, not a label, see ASSM manual 2.4. Unfortunately, we've goofed already. The assembler doesn't know where CXBUF is; it must be told somehow that it is at BCCØ. The PTDOS system disk contains two files, PTDEFS and NPTDEFS, that supply this information in a form the assembler understands. Since our source file has no line numbers, we'll use the one without numbers, NPTDEFS, and tell the assembler to read that file to obtain the necessary definitions. We do this by adding the statement " COPY NPTDEFS" to the source file. We could have skipped the COPY statement and changed the ORG statement to " ORG ØBCCØH" (Note that "ORG BCCØ" contains three distinct errors-- no space before ORG, BCCØ will be treated as a label not a number because it starts with a letter, and if it were recognized as a number it would be assumed to be a decimal one because it has no "H" at the end). It is better to do the COPYing (which will be needed later anyway, as is usually the case) and use CXBUF, which may mean something to the reader, than to fill the source code with numbers like ØBCCØH that are hard to remember and easy to confuse.

Now that the assembler will know where to load our program, we must tell it that the program is to be run once loaded, and where in memory will be the first instruction to be executed. As is commonly but not always the case, execution should start with the first byte, which will be the one at CXBUF. We could write " XEQ CXBUF"; it is better to substitute the statement " XEQ $", where "$" means "the memory location of the next byte of code." The reason is that we may someday change the ORG statement to load the program elsewhere; use of the "$" in the XEQ statement will avoid the need to change the XEQ statement too in order to specify that the run address is the same as the loading address. Whichever form is used, this statement is essential for this program; if it is omitted the program will assemble properly, load properly when its name is typed, and then be ignored, and the whole exercise will accomplish nothing.

At last we're ready for the actual instructions to do what we want--put a ØØ at address A89Ø. These require mnemonics; one set might be: MVI A,Ø <move a Ø to the A register> and STA ØA89ØH <store the contents of the A register at A89Ø.> It's better practice to give the memory location, A89Ø, a symbolic name, let's say "atpro" for "attribute protect", then refer to it as such in the instruction. To do this, we define the label by an EQU statement, "atpro EQU ØA89ØH", and rewrite the store instruction as " STA atpro." This may help us remember the significance of address A89Ø; in addition it will simplify changing the program if A89Ø turns out to be the wrong address, as one change in the EQU statement will effect the necessary substitution throughout the program however long and however many references to the same memory location it contains. And, one of the assembly-language tricks that everyone knows is to use " XRA A" instead of " MVI A,Ø"; the former does the job using one less byte of memory and almost twice as fast.

Having done all we wanted, what next? We can't just end the program there, because when it finishes running the computer would look for its next instruction at the next memory location, the contents of which are undefined. If this program were part of a larger program, or designed to be CALLed from BASIC or the like, we would end it with a RET instruction, directing return to the calling program. However, since the program is to be run by typing its name, we have nowhere to return to, except the Command Interpreter. To figure out how to return to that, we have to struggle through Sections 6 and 7 of the PTDOS manual. Section 7 explains the four types of returns possible--ABTOP for serious errors, RESOP to reset the system, RTROP to set a trap, whatever that means, and RETOP to return normally. We want RETOP, the normal return, mostly because the use of RESOP (or the equivalent, SRESET) would shift control back to the keyboard at the end of our program willy-nilly; we don't want this to happen if our program is being executed in a macro through DO or SETIN, because we want the rest of the macro (typically REATR) to be executed too.

Section 6.4.1 gives the calling sequence--e.g., assembly instructions needed--in this case " CALL SYS";" DB RETOP". We find on reading about RETOP on page 7-16 that we don't have to load any registers before the " CALL SYS", and that there are no error returns from the call, which means we don't have to worry about the JMP <error routine> mentioned in Section 6.4.1.

Cont. Pg. 3.

Cont. from Pg. 2.

The manual doesn't say so specifically, but we don't have to concern ourselves with the "normal return point" mentioned in Section 6.4.1 either; the whole point of the system call is to leave our program never to return. The definitions of SYS and RETOP are in NPTDEFS, so the assembler will obtain them through the COPY statement.

Here's the complete source code, with the pseudo-ops slightly rearranged for clarity and with appropriate remarks:

```
*        ZAP:S -- Source code for ZAP, a program to defeat
*        PTDOS attribute protection
*
       COPY NPTDEFS    copy the definitions
atpro  EQU ØA89ØH the key memory location (A8B1 for 1.4)
*
       ORG CXBUF      command execution buffer
       XEQ $          start at the top
*
       XRA A          make A hold Ø
       STA atpro      put it in atpro
*
       CALL SYS       and go home
       DB RETOP
*
```

And with that on the screen, less the remarks if you insist, we're ready to hit Control-F to exit EDIT, or otherwise to write the program to the file ZAP:S.

Step 2 - Assembling the Code

Now that we have a file containing what we hope is the correct source code, we must assemble it into a series of machine instructions, like the unintelligible string of CD's and 23's that appears when you use the DUMP command. This is the function of the assembler, ASSM. It reads our file of source code and writes to as many as four other PTDOS files. Unless one of them is a device or the console output file, ASSM will do its job with much clanking from the Helios but no visible results. As explained in the ASSM manual, it is invoked by the command ASSM source filename, {list filename}, {object filename}, {error filename}, {cross reference filename}, {options}.

Naturally the source file must exist and be specified; it must also contain valid source code or various assembly errors will result and be reported to the error file. A listing is essential for later debugging; the listing file may be a device if desired. When a source file has been only slightly modified and it seems likely other modifications will occur, it will save time to omit the listing file; a listing may always be created later by running ASSM again.

The object file, in our case ZAP, will contain the actual code to be loaded and executed, and must be specified if any executable program is to result. Errors will be written to the console output file if no error file is specified, which for most short programs is satisfactory. Don't, however, direct that both the listing and errors be sent to the screen for a program of any length--the listing will speed by so fast you'll miss any errors. A cross-reference file is of use primarily for long programs, and may be created faster by the PTDOS XREF command; the options are fairly useless and don't work reliably anyway, at least on my system.

So, let's assemble our program to ZAP, simultaneously listing it to ZAP:L, sending errors to the console and omitting the cross-references: ASSM ZAP:S,ZAP:L,ZAP <cr>. If no errors occur, ZAP is now assembled. A look at ZAP:L reveals that the instructions are only eight bytes in all, from the AF for XRA A to the ØD for RETOP. However, it is equally significant that ZAP (DUMP it to see) is six bytes longer--the assembler puts it in image format, as described in Section 5.7.3 of the PTDOS manual, and includes the byte count, load address and starting address in the file.

Step 3 - Running the Program

The object file, ZAP, is an image file, type I., block size 1ØØH. To run it, either type "ZAP <cr>" after the PTDOS prompt or include the name ZAP in START.UP or another macro and execute it with DO or SETIN. Once ZAP is assembled and runs properly, the source file, ZAP:S, is of purely historic interest--but don't KILL it yet. Like all image files, ZAP is a string of symbols almost untelligible to humans that resists direct editing. Most of the time, making a change to ZAP requires locating the source file ZAP:S, changing it through EDIT or whatever, reassembling it as changed through ASSM, and then running the new object file, and doing the whole process over and over again until it's right. This is slow work, and a good part of the reason why writing a long assembly program takes a lot of time. As with any hobby, some of us get hooked and don't mind the time spent, and you may well too.

Even if you decide to stick mostly to BASIC, a little time spent with assembly programming will allow you to write handy little routines like ZAP and will, perhaps more importantly, significantly increase your understanding of PTDOS and the SOLOS monitor and increase their usefulness to you. Here's an easy project to start with: Write and assemble UNZAP to restore the contents of A89Ø to 1ØH when you are finished reattributing a protected file. Two hints-- EDIT may be used to edit from one file to another, changing it while preserving the original, and the statement MVI A,1ØH should be in your new source code. Good luck!

Dear Tony,

I offer for publication in Proteus/News "yet another" set of CP/M user area routines for the SOL as well as notes on the A/J 841 and a patch for the ELECTRIC PENCIL. SOLUSER9 started with the minimal routines required for the SOL (about 75 bytes including a printer driver as well as the <CR> trap) and includes all of the "bells and whistles" one could desire without coming into conflict with CP/M application programs.

HELP!!! Following Stan's comments in Joe Maguire's review of the Digital Deli N*S software (P/N 2(3):11), I sent a note of inquiry to John Dvorak including a self-addressed stamped envelope for reply. Since then, I have heard nothing. If there is a mail-order source for this software, please publish a note giving the address. Technology Systems (208 Greenwood Ave., Bethel,CN 06801) has been working on a user-relocatible version of N*S DOS and BASIC as well as fixing some errors that existed in releases 4 and 5. One of their objectives is to be able to begin the DOS at F000H, followed by BASIC at FD00H, thus adding 12K to the space available for BASIC application programs. This software should be available by now.

With regards,

Jim Byram

→ Revised ← Please read carefully

Proteus Cassette Software Library January, 1980
Operating Policies and Procedures

PURPOSE: The Proteus Cassette Software Library exists to serve the Proteus membership. The library accepts contributions of software from members, and makes it available to the members at an attractive price. As such, it is a dynamic entity; it must grow to survive. As an incentive to contributors, the library software is offered at a discount to those who have contributed programs or data files.

PURCHASES: Current library cassettes are priced at $18.00 each. This price includes a quality cassette in a hard plastic case, packaging in a protective envelope, and postage. All cassettes will be sent via First Class Mail. All orders must be prepaid. Checks and money orders are accepted; send cash at your own risk. Payment must be made in U.S. dollars; foriegn purchasers except for Canada and Mexico, add $1.00 for air mail shipment.

To encourage support of the library, a credit of $10.00 is allowed for the donation of acceptable software to the library. Thus, the net price is $8.00 per cassette, with discount.

CONTRIBUTIONS: The library is constantly seeking new software. However, we are able to accept software for distribution only if it is in the public domain, or with the permission of the author or copyright holder. Programs copied from magazines and books are generally not acceptable, since they are copyrighted by the author or publisher. Similarly, propriatary programs can not be accepted. Remember, though, that you cannot copyright an idea, or an algorithym; just the specific way of expressing it. So, it is possible to use the ideas of a published program to write a similar one of your own. The contribution of an original program or data base earns you a purchase discount credit.

Certain magazines, notably Dr. Dobbs Journal and Recreational Computing, give blanket permission to use their programs, so long as proper credit is given. Programs from these sources are acceptable, provided that you have entered the necesary credit lines. In this case, you are donating your time and effort in keying in the program, rather than the program itself; you still get a discount credit.

In summary, then, for your program to be accepted you must send a signed statement that you know the program to be in the public domain OR that the program is your original creation and that you give the library permission to distribute it OR a letter from the author or copyright holder giving the library permission to distribute OR a program from a publication which gives permission to reprint. Unless one of these conditions are met, the program can not be used, and a credit will not be issued. Note that the library does not insist that your rights to a program be signed over to us; all we ask is your permission to publish and distribute it.

UNLESS SPECIFICALLY REQUESTED OTHERWISE, your cassette itself will be a part of your contribution. Except for the most expensive cassettes, it costs the library more in time and money to keep up with and return the cassette than the cassette is worth. Also, the library needs scratch cassettes to help prepare the distribution masters. Radio Shack's Realistic Low Noise - High Frequency cassettes are cheap and work well. Contributions can be made at any time. The library will send you an acknowledgement of your contribution, and a discount credit which you can use on a future cassette purchase. Or, you can send your contribution with your order.

CONTRIBUTION FORMATS: At the present time, the library is accepting donations in the following formats:

1. PTCo BASIC5 internal compiled format, i.e. SAVE name
2. PTCo ECBASIC internal compiled format, i.e. SAVE name,C
3. PTCo ECBASIC text format, i.e. SAVE name,T
4. Assembler source files as memory images of ALS-8/SOFT1 internal format in block access tape files. Each source line consists of:
    A. The length byte, which gives the length of the entire line, including the length byte and the trailing <CR>,
    B. A four-digit ASCII line number, including leading 0's,
    C. A space (between the number and the text),
    D. ASCII text source line, and
    E. A carriage-return character, 0DH.

   This file format is written by the SAVE command of the newer versions of SOFT1, and by using SOLOS/CUTER to save a memory image of an ALS-8 file or an older SOFT1 file.
5. Assembler source files and BASIC programs as ASCII text in the PROTEUS STANDARD BYTE ACCESS tape format, described below. The PROTEUS STANDARD BYTE ACCESS format is intended to be the universal bridge between tape-based systems and the various disk systems.
6. Executable object code, designed to work under SOLOS/CUTER. Where possible, SOLOS/CUTER should be accessed only through the entry jump table. If necessary to access by calls to internal routines, these should be documented so that users can adapt if necessary.
7. CP/M programs and text files in the block access tape format of the DISKTAPE program written by Richard Greenlaw, and distributed on library cassettes C10 and C11. So many of our members are now using one of the many varieties of CP/M that we want to offer this as a contribution format. However, most of the programs will be converted to one of the other formats before distribution so that they can be used by all. An exception will be programs which make use of the CP/M disk facilities; others couldn't use them anyway.
8. CP/M files on 8" single density soft-sector diskettes. Again, all applicable programs will be transferred off the disk and distributed on cassette so that all can use them.

   CP/M programs and files should have specific application to SOLs or our compatable computers; we don't want to go into competition with the CP/M users group.

PROPOSED STANDARD FOR BYTE ACCESS TAPE FILE EXCHANGE: When our User's Society was first formed, most of us were using the SOL/CUTS block access tape format for most of our programs and data. Because of this common link between all SOL owners (and compatable S-100 systems operating under CUTER), software transfer was simple and easy. However, at this time (early 1980), most of us are using some kind of disk system, and because of this, block access tape is no longer a universal exchange medium. However, we can still use the byte access tape format. Unfortunately, the various disk systems and other operating systems do not necessarily use a compatable internal format. To help overcome this obstacle, the library is proposing the PROTEUS STANDARD BYTE ACCESS FORMAT. The idea is that drivers can be written to allow the various operating systems to write a byte access tape file in a universal format, and that other drivers can be written to read in this format. The format which is being proposed is identical to that used by the user at the keyboard; thus the operating system sees the byte access tape file as a "fast typist".

Cont Pg. 5

Cont from Pg. 4

The WRITE drivers should write the STANDARD BYTE ACCESS tape files in this format:

    <line#> <ASCII text> <carriage-return>

The WRITE driver may operate as a Custom Output routine under SOLOS/CUTER, or it might stand alone or be a part of another program. Normally, it would intercept a listing of the program, and send it to the tape instead of the screen. The WRITE driver should filter out leading spaces (but not leading zeros), line-feeds, nulls, and other control characters. One of the first user efforts at a WRITE driver for this format was written by Richard Greenlaw to convert BASIC5 programs to ECBASIC, and was published in Solus News, V1#3.

Similarly, the READ drivers would feed the tape characters to the operating system when it requests keyboard characters. The READ driver should be able to accept as input the WRITE format specified above without requiring any additional characters. the READ driver should be able to tolerate nulls, line-feeds, and leading spaces in case the WRITE driver did not do its job properly. If the operating system requires length bytes, linefeeds, or other special characters, then the driver or the system should provide them. This would usually be done by the system automatically if the READ driver properly simulates the "fast typist". The READ driver should handle end-of-file and tape-read-error conditions, typically by giving a message and switching back to the keyboard for further input.

This format is compatable with the byte access format used by ECBASIC, PTCo EDIT, and the PTCo PACK and UNPACK utilities. Currently, the drivers are available to interface the PROTEUS STANDARD BYTE ACCESS format with the following systems:

1. Northstar Disk	TAPEDOS by Joe Cumming
2. Micropolis Disk	Interfaced by Melvin Dalton
3. CP/M (all disk formats)	BYTE patch by Lewis Moseley, Jr.
4. ALS-8	READ driver as a part of the ALS-8-X extensions package.
5. General	General READ and WRITE drivers for any program which does I/O through SINP and SOUT, by Lewis Moseley, Jr., building on the work of Richard Greenlaw and Joe Cumming.

All of these are in the Cassette Library. The CP/M driver is on C10 with the other CP/M software. All of the others are on C9. The library welcomes comments, suggestions, and especially compatable drivers and software.

DOCUMENTATION: All programs should be fully documented. Where at all possible, the documentation should be on the magnetic media so that the library, and the members, will not have to bear the cost of duplicating hard copy documentation. For BASIC programs, the documentation should be in the form of imbedded REMARK statements and/or user prompts in PRINT statements. For object programs, the documentation should be in the form of well commented source code with instructions. If you do not wish to make the source public, then separate documentation should be provided or the program should be self prompting. As a last resort, we will accept hard copy documentation.

COPYRIGHTS: To the extent possible, all Proteus Cassette Software library programs are Copyrighted. This is done solely to prevent commercial exploitation. Programs which are shown as copyrighted by the individual authors, or otherwise explicitly marked as having restricted reproduction rights, may not be reprinted or copied without the permission of the author. All other programs may be copied for any NON-COMMERCIAL purpose, provided that proper credit is given to Proteus. For information concerning reproduction for commercial purposes, contact the Proteus main office at 1690 Woodside Rd., Suite 219, Redwood City, CA 94061.

Thus, except for the restricted programs, a club or local group can buy a single copy and share it among its members. (We know you do it anyway, we're just making it legal.) To make it even easier, a copy of TCOPY, the CUTS tape copy/verify utility by Lewis Moseley, Jr., will be included on most current and new cassettes.

WARRANTY: All programs are recorded redundantly on the cassettes. We guarantee that you will be able to read at least one copy of each program. If you cannot, return the tape and it will be replaced free of charge. (We recommend that you copy the files to a working tape and keep the library tape as a backup.) As the big companies say, this warranty is in lieu of all other warrienties, express and implied.

All of our software is donated. Most donors are reliable; some may not be. You must determine whether a particular program fits your needs. Never trust important data to an untried program, or make decisions based on one.

CORRESPONDENCE: Address all inquiries, orders, contributions, and other correspondence to:

Proteus Cassette Software Library
c/o Lewis Moseley, Jr., Librarian
2576 Glendale Ct., NE
Conyers, GA 30208

If you wish a personal reply, please enclose a self-addressed stamped envelope (SASE). A catalog showing the contents of all of the library cassettes is available now. If you want a copy, send a SASE. This information has been and/or will be published in PROTEUS NEWS.

INTERNATIONAL NORTH STAR USERS ASSOCIATION FORMS

After considerable discussions with North Star Computer Company a users association (similiar to SOLUS/PROTEUS) has taken steps to form. The aims and goals of this association are to disseminate information to all North Star equipment owners; to act as a buffer for customers with questions regarding problems or for those needing information and North Star Computer Company.

The association will be totaly independant of North Star Computer Co. and will derive its operating expenses through dues and a newsletter which will be published quarterly.

Dues have been set at $15 per year and members receive the newsletter free. This group is a long time comming and needs all the help they can get. Please send in your dues as soon as possible and send in any articles, programs, etc. for publication.

All those interested should contact:
N.S.U.A.
131 Highland Ave
Vacaville, Ca 95688

More Notes on the PTDOS Parameter Scanner (PSCAN)
by Joe Maguire

Larry McDavid's article on the PTDOS parameter scanner (Proteus News Vol 2 #5) was informative and accurate. Lately I have had extensive practice with the PSCAN routine while writing a machine language routine to salvage data from messed up Helios disks. (To be placed in the Helios library when completed) Larry's explanation helped clear up the sometimes confusing discription in the PTDOS manual. Below is a diagram I have found useful in explaining the operation of PSCAN to others.

```
        A      |       B        |     C
               |                |
    *BIGPROG   | /1,A700,256:D  | ;NEXTPROG
               |                |
               |                |
```

Fig 1.

Figure 1. represents a command line as typed to PTDOS. The prompt character (star) is shown at the left followed by the name of the file to be loaded (BIGPROG) followed by arguments. The operation of PSCAN on this line is as follows:

PTDOS contains a routine called the Command Interpreter. The CI is responsible for taking what the user types and passing it to other PTDOS routines to be acted upon. The CI relies on PSCAN to take the first look at what the user typed in. The PSCAN call contained within the CI begins reading the above line from left to right starting with the character next to the prompt and continuing until it finds a special character whichs tells it to stop reading. The special character is called a delimiter and in this example it is a space. Other characters can be used as delimiters which cause different things to happen within PTDOS. A space delimiter tells PTDOS to load the file and begin executing it. (The file must contain the information PTDOS needs to know to do this correctly) Assuming that our BIGPROGram has loaded and is now running, we move to part B of the command line.

Our super program needs to get some information from the user in order to do the teriffic job we want it to do. How does it get it? Why our old friend PSCAN of course! BIGPROG now makes a call to PSCAN to pick up the next part of the command line and act upon the information. There are a few important things to point out about how PSCAN operates on part A and part B, and later part C, of the command line.

In part A, PSCAN was called by the CI. In part B, PSCAN is called from BIGPROG. Just how the characters returned by PSCAN will be acted upon is determined by the programmer. Go back and read Larry's article for an example of how he used the comma, semicolon, carriage return, etc. to cause various actions within his program. By agreement, the semicolon has been designated to mean that we now want to pass control back to PTDOS. When BIGPROG receives the semicolon from PSCAN it will pass control back to PTDOS and the CI which will now read and load NEXTPROG. BIGPROG does that because we wrote it to do so.

In summary, PSCAN can be thought of as a vacuum cleaner with two output chutes. It starts going down the command line sucking up the characters one by one and directing them out the chutes. Letters and numbers go out one chute and delimiters go out the other. (for a list of the delimiters see the PTDOS manual) The user written routine that called PSCAN "catches" the characters and acts upon them. PSCAN can also do other things for you such as conversion from Ascii to binary. PSCAN allows input to a program to be accomplished automatically, unattended by the operator. By contrast, using the Console Input routine (CONIN) requires that execution stop and wait for the user to type something on the keyboard.

As a footnote, I have just received my copy of CP/M 2.0. I had great expectations for this update but my initial reaction is that it still falls far short of PTDOS. The more I become familiar with PTDOS routines the more I realize the tremendous power availabe to the programmer. If only PTC had adopted the IBM format at the outset, I feel certain that PTDOS would have become the standard.

THE LATE LAMENTED

WITH DUE RESPECT TO JOE MCGUIRE, I BELIEVE HIS ASSUMPTIONS REGARDING THE DEMISE OF PROCESSOR TECHNOLOGY MISS THE ESSENTIAL POINT. I AM WELL AWARE OF THE PROBLEMS THAT P.T. HAD WITH THEIR DEALERS; I KNOW SEVERAL QUITE WELL. BUT I DO NOT BELIEVE THIS WAS THE ULTIMATE CAUSE OF THE EARLY DEATH WE ALL REGRET SO MUCH.

THERE IS IN THIS COUNTRY A NUMBER OF DEDICATED AMATEURS, SOMETIMES CALLED HOBBIESTS, BETTER DESCRIBED AS PIONEERS, AT THE FRONT OF EVERY NEW TECHNOLOGY. YOU FIND THEM IN RADIO, FLYING, AUTO CLUBS, ETC. THEY ARE FOR THE MOST PART SUBSTANTIAL MIDDLE CLASS TYPES WITH THE EARNING POWER NECCESSARY TO BUY AND MASTER WHATEVER IS NEW AND CHALLENGING. WE IN PROTEUS, I SUSPECT, ARE LARGELY IN THAT GROUP, BUT TOO FEW IN NUMBER AND OH SO FICKLE.

SOMETIME IN THE EARLY PART OF 1978 THE FLOW OF MONEY FROM AMATEURS BEGAN TO DRY UP. FORTUNATELY FOR A FEW MICRO-COMPUTER MFG. THERE WAS A SECOND, LARGER, AND EVEN BETTER HEELED GROUP JUST BECOMING AWARE OF THE POWER OF SMALL COMPUTERS. A GROUP THAT IN NO WAY RESEMBLE THE HOBBY CROWD. THEY ARE OF COURSE THE SMALL-BUSINESS PEOPLE. A FEW HAD STARTED IN COMPUTING WITH THE HACKERS, BUT MOST WERE NOT LOOKING FOR FUN AND GAMES OR ANY SORT OF CHALLENGE. THEY WERE LOOKING FOR HELP, FOR A TOOL THAT WOULD MAKE LIFE A LITTLE EASIER AND HOPEFULLY A LITTLE RICHER.

THE MARVEL OF THE TECHNOLOGY, THE LOVELY WIRES AND CHIPS, THE NEW JARGON TO LEARN, MEAN NOTHING TO THEM. WHAT WOULD LURE THEIR MONEY OUT OF THE BANK WAS WHAT THE THING WOULD DO FOR THEM AND THEIR BUSINESS AND HOW EASILY IT WOULD DO IT. IN A WORD, APPLICATION SOFTWARE. TURNKEY IS NOT A DIRTY WORD IN THIS GROUP OF PRACTICAL USERS. ECSTASY IS A WORKDAY WITHOUT PROBLEMS. AND RELIABILITY IS THE NICEST THING YOU CAN SAY ABOUT ANYTHING.

A TOOL THAT MAKES NO WAVES, THAT REQUIRES NO LONG LEARNING PERIOD, THAT DOES USEFUL WORK, IS WHAT INTERESTS HIM ABOVE ALL. BUT PROC. TECH. HAD NO APPLICATION SOFTWARE TO SELL. WORD-WIZARD WAS JUST AVAILABLE AND WORD-PROCESSING WAS NOT A HOUSEHOLD WORD AS YET. CERTAINLY HE ISNT GOING TO PAY 6000 BUCKS FOR A FANCY TYPEWRITER. FORGET ABOUT PROGRAMMING. ANYTHING THAT TAKES SO LONG TO LEARN AND DOES NOT DO A HECK OF A LOT ANYWAY IS NOT ON HIS LIST OF FAVORITE WAYS TO SPEND THE EVENINGS AND WEEK-ENDS. AND NO PROS, UNDERSTANDABLY, WERE WRITING SOFTWARE FOR A BASTARD DOS THAT PROMISED LIMITED SALES. NOT WITH THE TRS-80 EVERYWHERE AND BEGGING FOR ANYTHING THAT WOULD RUN. AND PET AND APPLE WERE YET TO BE EXPLOITED. SO WE WAITED IN VAIN. AND MOST DIDNT GIVE A LOOK AT SYSTEMS THAT WERE AT BEST PROMISES.

SO THE CASH FLOW SLOWED AND CREDITORS, BURNED OR SKITISH BY THE COLLAPSE OF IMSAI, POLY, ALTAIR AND THE OTHER EARLY DROP-OUTS, WERE ASKING FOR CASH IN ADVANCE, A COMMON BUSINESS HEDGE IN SUCH OMINOUS SITUATIONS. SO IT MUST HAVE BEEN OBVIOUS EVEN TO SUCH BAD BUSINESS HEADS AS THE BRASS AT P.T. THAT THE FUTURE BELONGED TO THE LIKES OF RADIO-SHACK, PET AND APPLE.

LETS NOT FAULT THEM FOR SAVING THEIR INDIVIDUAL REPUTATIONS BY AVOIDING BANKRUPTCY AND LAWSUITS BY SIMPLY SHUTTING THE DOOR. I USED TO THINK THAT THEY OWED ME SOMETHING FOR SUPPORTING THEM IN THE EARLY DAYS, BUT THATS NONSENSE. WE ALL GAMBLE WHEN WE GO INTO SOMETHING NEW. I DON'T FEEL NOW THAT WE WERE EXPLOITED; WE GOT OUR DOLLARS WORTH IN QUALITY OF MERCHANDISE AND IN BEING IN AT THE BEGINNING OF SOMETHING IMPORTANT AND EXCITING. FEW OF US WILL EVER GET THE MAXIMUM OUT OF OUR LITTLE COMPUTERS.

AS TO THE DEALERSHIP PROBLEMS THAT P.T. BROUGHT ON ITSELF, A CONSIDERATION BUT NO MORE THAN THAT. SEEN AN IBM, WANG, OR SPERRY-UNIVAC STORE LATELY ? THE MARKET WILL SEEK OUT THE BEST. HAD THEY EXPLOITED THEIR EARLY LEAD, STUCK TO THE STANDARDS AS BAD AS THEY WERE, AND GIVEN THEIR SOFTWARE DEVELOPEMENT TO THE SOFTWARE INDUSTRY (MICROSOFT, DIGITAL RES., ETC.) THEY MIGHT BE THE SMALL COMPUTER INDUSTRY LEADER TODAY, AND THE LEADER DOESNT NEED THE GOODWILL OF BYTE-SHOP, COMPUTER WORLD, OR ANY RETAIL STORE. GET YOUR PRODUCT INTO THE HANDS OF A FEW AND IF ITS THE SOLUTION TO A PROBLEM SUCCESS WILL BEAT THE DOOR DOWN.

Cont. Pg. 7.

Cont. from pg. 6

PROCESSOR TECHNOLOGY BUILT A BETTER MOUSE TRAP, BUT FORGOT THE CHEESE. WE, THE MICE, CAN DO SOMETHING ABOUT THAT. PROTEUS, THANKS TO STAN SOKOLOW, LEW MOSELY, FR. MCGAHEE, TONY SEVERA, AND OF COURSE JOE MCQUIRE AND A FEW MORE, IS AN AMERICAN INSTINCT, A KNEE-JERK IF YOU WILL. SURVIVORS, DOING FOR OURSELVES, BECAUSE THE ALTERNATIVE IS NOT IN OUR PHILOSOPHY, GETTING THE JOB DONE TOGETHER, DESPITE THE ODDS, THATS WHAT WE ARE.

IT WILL MEAN A LOT OF WORK FOR A FEW, BUT WHAT WE MUST DO, ALL OF US, IS CONTRIBUTE. NO MATTER HOW INCONSEQUENTIAL IT MAY SEEM, IF IT SOLVED A PROBLEM FOR YOU IT MAY DO THE SAME FOR SOME BROTHER. IF YOU HAVE A SOFTWARE SYSTEM THAT DOES A JOB, A NEW A GAME THAT DELIGHTS, A PROGRAM THAT INSTRUCTS, SHARE IT. IF IT IS A DUPLICATE, MAYBE ITS A LITTLE BETTER. IF ITS ROUGH AROUND THE EDGES SOMEONE WILL KNOW HOW TO SMOOTH IT OUT. IF ITS NOT YET COMPLETE SOMEONE WILL KNOW HOW TO ADD THAT FINAL TOUCH. WHAT IS IMPORTANT IS THAT TONY AND LEW AND THOSE THAT CAN HELP HAVE LOTS OF MATERIAL TO WORK WITH. AND THERE WILL ALWAYS BE THE HARDWARE PROBLEMS TO DEAL WITH. WE NEED SOME TUTORIALS ON WHAT GOES ON INSIDE THE THING. WE CAN PUT OFF BUYING SOFTWARE FOR A WHILE, BUT WHEN THE MACHINE IS SICK WERE OUT OF BUSINESS. THE COST OF THE SIMPLEST REPAIRS IS STAGGERING TODAY. SO LETS SHARE WHAT WE KNOW ABOUT KEEPING SOL ON ITS FEET, WHERE ONE SHOULD LOOK, WHAT ONE CAN OR CANNOT DO FOR HIMSELF, AND, LORD FORBID, IF IT COMES TO IT WHERE CAN WE GET HONEST SERVICE AT A REASONABLE PRICE.

ENOUGH! TOUCH ME ANYWHERE AND GET A TWENTY MINUTE LECTURE. WE REALLY CAN DO IT YOU KNOW. IF EVERYONE SHARES, WE CAN HAVE AS GOOD A LIBRARY AS THE CP/M USERS GROUP, BUT EVEN MORE EXTENSIVE BECAUSE WE CAN SHARE TAPES AS WELL AS DISKS. AND WE CAN ALL BE KNOWLEDGEABLE ABOUT OUR HARDWARE AND ITS MAINTAINANCE.

THANKS PROTEUS

EARL

5th West Coast Computer Faire Report

by Tony Severa

Well, the 5th West Coast Computer Faire is over and I would like to share the following reflections:

I was very pleased that so many PROTEUS members came by my booth to say hi. There didn't seem to be as many as there was last year, but, last year PROTEUS had its own booth. That booth we had last year, by the way, was GENEROUSLY DONATED by Mr. Jim Warren. I don't believe we ever thanked him in our newsletter, but forgot to. THANK YOU JIM FOR YOUR CONTINUED SUPPORT YOU HAVE SHOWN US THESE PAST SEVERAL YEARS!

Bob Burns dropped by to say hello and continue to give us his support. Those of you who were with SOLUS in the beginning remember Bob in that he was the person responsible for starting SOLUS in the first place. SOLUS was then transfered to Stan Sokolow who then renamed it PROTEUS. SOLUS and PROTEUS are the same only the names are different. I have noticed that some people are still confused about the relationship between SOLUS and PROTEUS. Hope this clarifies it some. Anyway, Bob Burns originaly organized and ran SOLUS and did all he could to develop a good relationship with Processor Tech. It was that relationship that allowed us to get copies of all the manuals and updates before they met their demise. Bob still works behind the scenes and supports SOLUS/PROTEUS in every way possible. THANK YOU, BOB!

H-4 RECIEVES A NEW FACE LIFT

After several H-4 library diskettes were sent out we found, with your help, many bugs with the contents of that disk. That was the first library disk I had ever worked on, and I did not have time to check each program for correctness. I also blundered by forgetting to put several programs on it. What happened was that I accidentally used my work-disk as my MASTER and copied over my real MASTER disk. So, several of those programs have gone back to their originators for rewriting or recopying.

A new H-4 disk is going to be released by the time this issue is sent to you. Again, I do not have time to CHECK each program that it works or does what it is supposed to. I consider myself a NOVICE and cannot understand half of the programs on it anyway. When you send in a program we expect and assume that it is reasonably bug free.

For those of you who ALREADY bought H-4, please return it with a three dollar check made out to 'Tony Severa' and I will diskcopy the new H-4 onto it and get it mailed back to you. The reason for the $3 is to pay for the diskcopy time and postage requirements. Please bear with me. The more I use the system, the more I learn and some day I'll be in a position of being able to check the programs more than I do already.

Tony

NOVICE-TO-NOVICE

Derived Functions and often needed values

When working with Basic, there is sometimes a need for a numeric value or a special function which is not part of Basic's standard offering. A good example of this is Pi. (3.1416...) What follows are a few tricks for obtaining some of these values from Basic itself rather than having to look them up.

Function or Value	Basic statement to obtain it
The value of Pi	Pl=4*ATN(1)
1.5707963 (used below)	N=Pl/2
The value of e	E=EXP(1)
To convert degrees to radians, multiply by R	R=Pl/180
To convert radians to degrees, multiply by D	D=180/Pl

(The following trig functions require their arguments in radians)

SECANT	of X	S=1/COS(X)
COSECANT	"	C=1/SIN(X)
COTANGENT	"	O=1/TAN(X)
INVERSE SINE	"	S1=ATN(X/SQR(1-X*X))
(The inverse sine of 1 = Pl/2)		
INVERSE COSINE	"	C1=-ATN(X/SQR(1-X*X))+1.5707963
(The inverse cosine of 1 = 0)		
HYPERBOLIC SINE	"	H1=(EXP(X)-EXP(-X))/2
HYPERBOLIC COSINE	"	H2=(EXP(X)+EXP(-X))/2
HYPERBOLIC TANGENT	"	H3=-EXP(-X)/(EXP(X)+EXP(-X))*2+1

Important note: Some of the functions given above can blow up with certain values of X. For example, the inverse sine or cosine will result in a divide by zero if X = 1. To get around this, test the value of X before performing the calculation.

contributed by Joe Maguire

NEWS FROM PROTEUS
by Stan Sokolow

SOURCE CODE PROGRESS REPORT:

I wish I could report that I have taken possession of the source code files from PTC, but not yet. If anyone wants a refund because of the delay, I'll be glad to send one, but I'm confident we're going to get the goods; two more weeks, I think. (Somehow this has a familiar ring to the old timers, doesn't it?)

PORTABLE PTDOS

Once we get the source code, an obvious thing to do is to modify it. PTDOS has the advantage over other operating systems in that it was tailored for the Sol, although it will work with any 8080/8085/Z80 computer, and it provides some very sophisticated features that many disk operating systems don't have. For example, it provides device independent I/O and the video editor is superb.

I know of a number of people who have access to time-sharing systems and many microcomputers (Intel, Zilog, CP/M systems, etc.) but they prefer working on their PTDOS system because it is so easy to use and flexible. The major complaints are that PTDOS is not ubiquitous as CP/M is, and that the Helios controller is the only one it works with.

I've been in contact with a number of people who used to work on PTDOS. I believe that I could put together a team to produce a new portable version of PTDOS, but I need to know how many people would be interested in having the new PTDOS for their system and what disks they want it to run on.

What I propose to create would be a new PTDOS that will have the following features.

1. It will be upward compatible from the present PTDOS, with perhaps a few minor changes needed. Your present programs in PTC Extended Disk BASIC will run unchanged, and your assembly language programs may need slight modification if any. Extended Cassette BASIC and BASIC/5 programs can be transcribed automatically into PTDOS files and run with little or no change. Your present copies of WordWizard, MailMaster, etc. will run unchanged.

2. It will operate on any mixture of disks and disk controllers, such as Helios, NorthStar, Micropolis, Micromation, ThinkerToys, Tarbell, Morrow hard disk, NorthStar hard disk, etc. Programs will not care which disk units they are dealing with. New disks can be incorporated into your system by simply copying the driver file from the library onto your system disk. A library of such disk drivers will be accumulated. Any mixture of sequential devices (printers, tape drives, modems, etc.) can be incorporated into your system in the same manner. No assembly language knowledge will be needed to do this, as long as the driver is in the library.

3. A CP/M emulator will be available. It will allow virtually all CP/M programs to run under PTDOS. A command will be supplied to move programs and files between the two disk formats. Also, a Solos emulator will let cassette programs run under the disk system.

4. It will take full advantage of the special features of Sol and VDM, but will also run with other video displays and terminals. Included in the base price will be the video editor, absolute assembler, vectored loader, disk BASIC/5, Extended Disk BASIC, and about 50 utility commands.

5. It will allow the Sol to be modified to move its built-in memory from C000 to F000 to allow full use of contiguous memory. PTDOS will be relocatable to the high end of your memory space.

6. Also available at reasonable prices will be a relocating macro assembler and loader, a video-oriented debugger (DDS--the best I've ever seen), an inexpensive COBOL, disk PILOT with video tape recorder control for computer aided instruction, a list-management language, PASCAL, FORTRAN, and a NorthStar BASIC to EDBASIC conversion program. Application programs developed by former PTC dealers will be available where possible, such as accounting, mailing list, sorting, horoscope casting, etc.

The price for the basic system will be similar to CP/M 2.0's price. Lifeboat Associates sells CP/M 2.0 for $170 to $375, depending upon the computer system. Our goal will be to match the low end, $170, but you will be getting more than CP/M gives at that price. The other major programs will be sold at reasonable prices.

Why bother with PTDOS? I've asked myself this question over and over. CP/M isn't great, but it's everywhere and has lots of application software. Here's how I see it.

Helios owners have a lot of programming time and get-acquainted time invested in PTDOS already. Some would like to add a hard disk (the 2-slot Helios cabinet has room for a hard disk inside it). Some would like to convert their Helios to a soft-sectored double-density controller without losing the ability to run PTDOS applications. I'd like to help them do it. I've kept my work on a replacement controller for the Helios on the back burner because of the problems of software portability.

NorthStar owners who still have only the NorthStar DOS really are missing the power of a full operating system. PTC's EDBASIC is similar to an enhanced version of NorthStar BASIC, so NorthStar owners with Sol's could easily move up to PTDOS.

Between these two groups (Sol/Helios, and Sol/NorthStar) we could have enough interest to create the new PTDOS, and it would fill a role that CP/M can't. Once we got that far, adding other disk systems would be easy.

Dealers have told me they feel there is a place for an operating system between CP/M and Oasis in complexity, and PTDOS has what they want. Many people who worked with both CP/M and PTDOS have told me they prefer PTDOS. But it needs more application programs and languages.

Considering all of this, I think it's worth trying to put together a portable PTDOS. But this is pure speculation on my part.

I could go on with ideas, but this is enough. Please give me some feedback and send a letter of intent if you are seriously interested in the portable PTDOS (or whatever we'll call it).

MAILMASTER, WORDWIZARD, ACCPAC ARE STILL AVAILABLE THROUGH PROTEUS

Proteus is now the U.S. distributor for the Basic Computer Group's application software which they wrote for Sol/Helios systems. We will sell them at the prices shown below, but dealers will receive a discount. Questions and orders should be sent to me, Stan Sokolow at the address below. Through the new arrangement, I will be able to sell the software and/or the manuals, with off-the-shelf availability.

	Full Price / Manual alone*
WordWizard	$300 / 35
MailMaster	$400 / 35
MailSort	$150 / 20
AccPac General Ledger and Financial Reporting	$600 / 35
AccPac Accts Receivable	$600 / 35
AccPac Accts Payable	$600 / 35

*Manual purchase will be credited to subsequent diskette purchase. Full price includes both manual and diskettes. Tax and shipping extra. COD okay. No credit cards.

A diskette with source or object code for additional printer drivers is available from Proteus with any BCG software purchase.

Stan Sokolow
Proteus
1690 Woodside Road, Suite 219
Redwood City, CA 94061

SOFTWARE WANTED

I am preparing to write a software routine for personal use and for contribution to the Proteus Cassette Software Library. This program will be an interface between the popular CP/M operating system and existing PTCo software. Since I am a firm believer in standing on each others shoulders, rather than on each others toes, I am asking anyone who has already attempted such a program to share his experiences with me.

The program would be called in from disk as a transient program by typing its name followed by the target program name, i.e.

```
    A> XEQ ECBASIC.BLK
```

The program would initially do the following five things:

1. Load and execute at the CP/M transient address of 1ØØH,
2. Relocate itself out of the way, perhaps in the C9ØØ range,
3. Find and read in at address Ø the specified disk file,
4. Load registers HL with the address of a surrogate jump table which would replace the usual SOLOS/CUTER jump table, and
5. Begin the program by a jump to address Ø.

The surrogate jump table should interface all, or at least some, of the SOLOS/CUTER functions to the CP/M system. The input routines (SINP and AINP) could simply jump to the appropriate points in the SOLOS/CUTER jump table. The output routines could send Pseudoport Ø to the CP/M console and all other to the CP/M lister. The Block Read and Block Write should locate and call in disk files rather than tape files. I suggest that the CP/M file-type .BLK be used for this purpose. A jump to RETRN should reboot CP/M. Eventually, sequential disk I/O routines could be written to simulate the SOLOS/CUTER byte access tape routines.

If you have inplemented all or part of a SOLOS/CP/M interface, please write to:

Proteus Cassette Software Library
c/o Lewis Moseley, Jr., Librarian
2576 Glendale Ct. NE
Conyers, Ga. 3Ø2Ø8

Thanks.

I recently aquired a double density North Star drive and controller and had to go through the pains of interfacing it with the Sol. It didnt use the same DOS as my old single density system, so it was time to gather my skills and phone lists and start calling. What came out of it was a nice little way of interfacing double density North Star DOS to the Sol. I share it so that no one else has to run up the phone bill I did trying to get it to work right.

Boot the disk drive.
Reset the Sol (upper-case,repeat)
Enter the following tables:

```
                          Jump Tables
EN 2ØØD: C3 ØØ 29
         C3 1Ø 29
         C3 2Ø 29
         C3 3Ø 29

                              I/O
EN 29ØØ: CD Ø5 29
         78 C9 B7
         CA 19 CØ
         C3 1C CØ

                              Keyboard Input I/O
EN 291Ø: CD 1F CØ
         CA 1Ø 29
         E6 7F FE
         7F CØ 3E
         5F C9

                         Clear Screen like Clear
EN 292Ø: C9

                   Control-C functions like Mode Select
EN 293Ø: CD 1F CØ
         FE Ø3 C8
         FE 8Ø C9
```

```
Type EXEC 2Ø28 (This should put you in N.S.DOS.)
Type LF DOS Ø
Type GO M2DØØ
Type MM 2ØØD,C D (return)
Type MM 29ØØ,1ØØ 8ØØ (return)
Type OS (will put you back in N.S.DOS)
Type SF DOS Ø
```

(Now you can boot up this disk and have full functions with North Star)

Tony

End-of-page Control for Pencil I

I have read several commentaries on Michael Shrayer Electric Pencil I. One of the main gripes has been the lack of the ability to program a stop at the end of a page in order to put a new sheet of paper into the printer. This is highly important to me since I use my system for quotations using bond letterhead paper.

I have a Sol 20, N* Disk, a Selectric Printer and Pencil I version SSN.

Since most Pencil I versions are very much alike, this fix will probably work on most of them.

Here is the fix: Make a memory dump starting at OB10 to OB3F which should get you in the general area. My version looks like this:

```
OB10  FE 0C CA CE 0A D2 AD 0A 3E 0A CD 41 09 3E 0D CD
OB20  41 09 3E 00 0E 00 0D F8 CD 41 09 C3 26 0B 3A 7D
OB30  22 32 7E 22 EB 2A 77 22 EB 22 77 22 CD 44 0B C8
```

Change OB29 from 41 to 38 and OB2C from 26 to 22.

```
OB28  CD 38 09
OB2B  C3 22 0B
```

Set PAGE LENGTH G1-72 to number of lines wanted. When printer stops, change paper, set top margin and use space bar to start printing new page.

I would be happy to give any additional help you may need.

I would be interested in receiving information regarding any other changes to Pencil I that would make it more useful or easier to use.

Elmer Tate
5513 Knollcrest Ct.
Dayton, Ohio 45429
(513) 434-5529

Dear Tony,

The latest PROTEUS (Jan/Feb,1980) just arrived, and I was most gratified to hear your ideas about dissemination of information, the Service Center,repair facilities, etc. First thing, let me add to the repair facility file. My Helios had been acting up and I found someone in this area who has the technical ability and the knowledge to deal with Sol and Helios.He not only solved the problem, but brought it up to the latest available revision. I called him today and urged him to join Proteus, which he intends to do,and to ask his permission to mention his repair service in Proteus.

Lee Mull
Vienna, Virginia
(703) 938-8219
Rates at present are $25.00 per hour and 20 cents/mile

He knows software as well as hardware and is able to work on a variety of disk drives.

Your admission that even you do not fully understand all the PTDOS made me feel better. Although I have had the system for quite some time,I never have made the fullest use of which it is capable. I also wish that more time were available to me to work with it. For example, I recall a note that appeared in PROTEUS sometime ago, asking how one might do away with the necessity of lifting off covers to change baud rate. I also wanted to know. I could print out a file or program at 300 baud by entering OUT P after the PTDOS asterisk, then loading BASIC, calling the program, entering a line SET OF= #1 (and of course another to SET OF= #0), and then running the program. Then if I later wanted to use WORD WIZARD, I had to switch rates to 1200 baud. At any rate after all this time it occurred to me to use the GET command to transfer the PRINTER Device file from WIZARD to my PTDOS work disk and then use SETOUT PRINTER and list FILES or PRINT. How good it was to have found the answer! Ah, but then also realized that this SETOUT command need not be used, but that PRINT O=PRINTER,<file> would also work. Then re-reading backissues of PROTEUS found Stan mentioned use of COPY. It also works -COPY <FILE>,PRINTER. Point is that while all this may be in the PTDOS manual, it really is not explained in a simple fashion for novices such as myself. Let us resurrect the old Novice corner to provide handy bits of information in order that others need not go through over the same old ground. Tutorials also were very helpful. Ah to have the knowledge of such giants as Maguire,Fr.McGahee,Mosely, not to mention Stan. There is something to be said for figuring it out on your own, but expert advice and suggestions saves covering a lot of useless ground. At any rate, I don't lift covers to change baud rates very often anymore!

Another idea - perhaps a get together and tutorial session of a week wherein some of the more knowledgeable could provide some'hands on'type instruction, and where we could meet one another on a national basis.

We even might arrange to have an annual PROTEUS conference,establish information networks via MODEMS or a variety of other projects. Could we take a poll, or get some feedback from other PROTEUS members who would be interested?

Frank J. Sanders
Towson, Md. 21204

Dear Tony Severa

Just read your editorial in my copy of Proteus/News for Jan/Feb. 1980 - You are wonderful - Your idea of helping the novice with the Sol, Helios is great - I am really one of those you referenced.

We bought our system about 4 years ago and I am still very much in the dark about all that it will do. We of course saw beautiful things being done at the dealership in Houston. They helped as much as they could from that distance but I am not what you would call a"rapid study" when it comes to computers. I have a tax business and we do some bookkeeping too - I have invested in a business program that now is working fine - it was written by Don Cooper now of FD Systems Inc. in Houston. He has been very helpful when I have had problems that a novice can't solve but again he is in Houston.

The Tax program I finally bought runs on a Crememco and at this writing we have the Crememco available for use here in our suite of offices. This will unfortunately no longer be true comes the end of April when we have to move because of lease termination.

Didn't mean to bend your ear so long but really wanted to tell you how I for one will be watching the Proteus/News for all the education I can gather.

Thanks again - you are a light at the end of a very dark tunnel for me.

Dorothy L. Dailey
Austin, Texas. 78747

Tony,

First, let me congradulate you on the assumption of the editor position of PROTEUS...I'm sure Stan needs the break after nearly three years of doing a super knockout job. The support of PROTEUS, now more than ever, will be of constant support knowing it is still in very capable hands and not about to "let down" Sol and Proc. Tech. equipment owners like the original company did! For that, I thank you, Stan, and everyone else associated with PROTEUS.

Secondly, as I mentioned to you on the phone, two work associates, Paul Adams and Bob Dietzel, along with myself, have formed a computer club in the Berks County/Reading Pa area. It is open to anyone sharing interests in computing from software to hardware, student to professional and is not limited by make of home computer. If anyone is interested, please contact us!

This brings me up to the final point...I need to find a Nortn Star single density, dual or single drive but with an S-100 controller card to expand to two drives...used but working. Please send the price you (PROTEUS readers) would like for your unit if you have one or phone after 4pm EDT @ 215-779-5018.

Ron Genova
601 Friedensbury Road
Reading, Pa. 19606

Dear Stan:

Enclosed is a check for 1980 dues and also another for Source Code items P12 and P15 which I would like to have sent on a Helios disk. I hesitated a little on the source code check because I don't understand assembly language source code very well yet. But, I realize how important this is to Proteus and Proteus is important to me. After I become proficient in Pascal, I'll tackle assembly language.

I have been trying some more to get the 64k version of Pascal going. I reassembled the interpreter and the RUN program. Now when I try to run a program, it seems to start but as soon as the ACK character appears, the bootload program is triggered and I wind up back in PTDOS. Do you know what might be causing this? Maybe PTDOS 1.5 behaves differently than PTDOS 1.4

Can you tell how the Pascal TRAP and EXIT extensions are used? With TRAP, is the first argument a memory location? With EXIT, what is the function of the interger argument? I wonder if TRAP could be used to call a routine to give random access files to SLAC Pascal.

In working with the SLAC Pascal, I noticed that the decimal ASCII codes have been shifted by 32 for some reason. For instance CHR(55) doesn't yield '7', instead you get 'W'. For CHR(n), n must be 32 less than the ASCII code of the character you want. The only reason I can think of for doing this is that it puts the control characters after the printable ones.

It has bothered me that when you provide keyboard input to a Pascal program, you don't see the characters on the screen as they are typed. While I haven't worked it all out yet, it appears that using GET(INPUT); C=INPUT@; rather than READ(C); is a way around this annoyance.

I hope to develop some functions and procedures to simulate the string handling extensions in UCSD Pascal.

I have discovered the function of the INITPATB mystery file on Helios disk H-1. Through communications with Sassan at SLAC, I have learned that they have the INITPATB command in the PTDOS START.UP file. Then, PTDOS will print the '_' character. Normally, PTDOS treats '_' the same as 'DEL'. If you want to print 'ABC_DEF', PTDOS will normally give you 'ABDEF' instead. But with INITPATB, you get what you want.

The problem is that INITPATB works with PTDOS 1.4 but not PTDOS 1.5. I obtained the source code form Sassan, but I don't know how to modify it for PTDOS 1.5. Maybe another Proteus member can help. The source code is as follows:

```
 TITL I/O INITIALIZER FOR NEW SLAC PTDOS DOS1.4.1 6/05/78
 COPY NPTDEFS
 ORG 0100H
 XEQ BEGIN
BEGIN MVI A,0 NOP
 STA 0B6CEH ZAP CONIN RZ WHICH SNAGS DELETES
 MVI A,7FH BACKSPACE CHAR
 STA 0B5ABH CONIN BACKSPACE
 STA 0B5F3H CONIN ECHOES FOR BS
 STA 0B6F7H VDM USES FOR BS
 CALL SYS
 DB RETOP
 END
```

When we get the Source to PTDOS 1.5 we'll be able to figure out the changes. -Stan

Speaking of mystery files, I still don't know what the "!" file is on Helios disk H-2.

(It's DISKCOPY. I got tired of typing so many characters when duplicating library disks. -Stan)

I'd like to see the idea of putting a hard disk inside the Helios box. I'm going to be working with WordWizard and about 15 megabytes worth of text soon. It would be nice to have it all in one place rather than on 60 or 70 floppies. (See page 8.—Stan)

A few weeks ago, I bought some of the $5 Processor Tech cassette tapes from Cheops Electronics. I'd like to transfer the games from Gamepac 1 & 2 to disk. When I tried, I got bad image files. Is there a secret to doing this?

You can add the Basic Computer Store in Akron Ohio to your list of former Processor Tech dealers still providing some support. I think they are the only dealership in the Midwest who will service a Helios. I haven't tried their service, but I do know that they are still willing to provide it. They have always been very cooperative when I have talked to them about other things. When I was in there, they told that they tried to contact Proteus (Solus) in the past but got no response. The man I was talking to was a little surprised to find out that Proteus is alive and well. Maybe you should send them a sample issue and some membership forms. Former dealers may be able to supply Proteus with the names of Sol owners who could be potential members. I had no idea what percentage of Sol owners are Proteus members. After talking with Tony Severa on the phone yesterday, it sounds like there is an awfully lot of potential out there for increasing membership. I hope the recent Proteus existance announcements in several magazines help.

Finally, a product recommendation for members. I recently bought some vinyl covers for my equipment from Cover Craft. You can find their illustrated ads in magazines including Microcomputing and Interface Age. Cover Craft offers made to fit covers for Sol (with and without monitor), Helios and SolPrinter (Diablo) as well as much other equipment. If what you need is not one of their stock items, they will custom make it to your dimensions. The Helios cover I got fits well and has a cutout for cables in the rear. The Sol with monitor cover encloses the front part of the monitor and the keyboard area of the Sol. It stops at the rear of the flat top area of the monitor to allow for antenna and cables. The SolPrinter cover extends down to the horizontal joint in the printer case. The bottom portion of the printer is left exposed. It is supposed to work with a forms tractor, but I haven't tried that yet. Cover Craft also custom made a cover for my cassette recorder. It looks like they lost the dimensions I sent with the order. It is much to large. I hope to send it back to try out their 100% Satisfaction Guarantee.

Cordially,

Michael A. McKelvey

RE: PROGRAMMING QUICKIES...LEWIS MOSELEY, JR.
PROTEUS VOL.2 NO.5

A NIFTY LITTLE TRICK FOR US SUPER-LAZY PEOPLE:
<CURSOR-UP> <LINE-FEED>
ACCOMPLISHES THE SAME AS:
<CURSOR-LEFT> <CURSOR-LEFT> <CURSOR-UP> <RETURN>
BUT WITH TWO LESS KEYSTROKES.

RE: SOFTWARE TECHNOLOGY MUSIC SYSTEM

SOFTWARE TECHNOLOGY'S MUSIC SYSTEM ALLOWS YOU TO ENTER A LINE #0000, BUT DOES NOT ALLOW YOU TO DELETE IT.
MAKE THIS SIMPLE PATCH TO PERMIT DELETEION OF LINE #0000:

```
      EN 00C7:
      00/
```

Ken Wong
Alberta, Canada

Arms
(or I'M stiken with you Joe)

With each issue of PROTEUS, I feel its value more. I look forward to finding nice little tidbits of information that have been contributed by my fellow Sol alumni that I can apply to my own system. This feeling has now risen to the level that I must now break my silence in hopes that I may contribute something useful to you, my fellow Solumnites.

First a little feedback. I am particularly greatful to Joe Maguire and all those of you who have shared your experiences, knowledge and desires with us in PROTEUS. I single out Joe because this is my attempt to respond to his article "A Call to Arms". Of tremendous interest to me also was the artical "COMMON SYMPTOMS OF FAILURE IN THE Sol" (unsigned). The "STACK CRASH" is one of my most common problems. (See my note to the right. — stan)

Now for a little background. I have a SOL/20-8KRA, SD Sales Expando Ram - 32k, 2 North Star disks drives (which I bought from Joe Maguire when I lived in Japan) and an IBM Selectric 735 printer. All of which I use for my own and my family's personal pleasure and education. Well, thats not quite true. I find the Electric Pencil very useful at work to write operating procedures, but it's still my pleasure.

A little contribution is now in order. Lewis Moseley, Jr. contributed an interesting programming quicky to cut a second or back-up tape after the original cut by hitting (cursor - left), (cursor - up), (return). A modified version of that is (cursor - up), (line feed) and that's all there is to it.

I mentioned that I have a printer. I have a software driver that sits at C900 and in order to use the printer I have to (SE CO C900)(RETURN)(SE O=3)(RETURN). Now in BASIC it's not always desirable to do this because I may only want certain things to be printed. The above commands can be accomplished in BASIC by placing the following instructions just before those things you want printed:

```
90  REM* SE CO C900       For NS* BASIC USE
100 POKE 51202,00         FILL instead of POKE
110 POKE 51203,201
115 REM* SE O=3
120 POKE 51207,03
```

After the printer is finished and I want control returned to the screen, I use:

```
200 FILL 51207,00
```

51202 is the user defined port at C802
51207 is the current output psuedo port at C807
These decimal representations of the hex addresses can be found in the back of the Extended Cassette BASIC User's Manual by finding 0C 000, which is 49152 (on page A5-1), then find the number (on page A5-4) in the far left-hand column that corresponds to 800 (in C800) and go over to the 2nd row of numbers for decimal number representing 802H, which is 2050 and add that number to 49152 to = 51202.

I found a patch in DR. DOBB'S for the North Star disk that provides double column listings for the LI command. The patch doesn't work with Sol for various reasons so, liking the idea, I set about to write my own patch. I hope you can use it.

```
     For Rel. 4
beginning addr.   old code           new code
     28F8         10                  20
     2601         FE 01 C2 11 26     FE 09 D2 11 26
     2611         CD 0C 27           CD EE 29
     29EE         GARBAGE            0E 07 06 20 CD 19 C0
cont 29F5            "               0D C2 F2 29 C9
```

There are a few things I would like to see in PROTEUS that some of you may know. I received a letter from Ralph Paulson, a former customer relations man with PTC, (in response to an inquiry I made) and he told me that in the next issue of ACCESS (which I never received) would be an article on how to replace the 21L02 memory chips on the Sol mother board with the 4114 8K chips that would service hi-memory, presumably from D000H on up. If anybody knows how to do it, I would like to see it.

I presently live in Dugway, Utah, and have been cut off from the small computer world since leaving Japan. Is there anybody in Utah besides me that has a Sol?

I find less and less in KILOBAUD and DR.DOBBS that applies to my interests. While they are good magazines I hope that more of the Sollumnites will contribute whatever they can to help PROTEUS prosper. Thank goodness for the few who seem to carry the load. I hope we don't burn them out.

Jim Gibson
5390A Carr Circle
Dugway, Ut. 84022

The "Common symptoms..." came from the Sol Technician's manual. I plan to put out a multi-volume set of books all about Sol, including the unpublished techinical chapters. Should be ready by the next issue. — Stan

---

Dear Stan,

I have a few questions that I hope you or other PROTEUS members might be able to help me with.

1. Is anyone working on getting the SLAC Pascal running on a Sol cassette system? If so how much memory will it take?

(ED. NOTE:
I HAVE THOUGHT ABOUT IT AND WOULD LIKE TO DO IT, BUT I'M JUST UNABLE TO FIND THE TIME. I MAY GET TO IT SOMETIME IN 1980. IT WILL NEED ABOUT 56K MINUMUM, WHICH IS THE EASY PART. THE TOUGH PART IS THAT IT WILL TAKE ABOUT 30 MINUTES TO EDIT, COMPILE AND LOAD EVEN A MINIMUM SIZE PROGRAM. DEBUGGING WILL THEREFORE BE TEDIOUS. I KNOW OF NO ONE ELSE WHO EVEN IS THINKING ABOUT DOING IT. --STAN)

2. Is anyone working on getting USCD Pascal or the "Tiny Pascal" written up in Byte running on a Sol cassette system? I have gotten the Byte Nyybles listing of the 8080 assembler language version of the "Tiny Pascal" and am starting to try to get it running on my Sol. However it will be a long process I think since I only have cassette and do not have a printer. I would be interested in hearing from anyone else who might be working on the same or similiar projects. If I succeed in getting the "Tiny Pascal" running I will attempt to get an O.K. from Byte to make it available to PROTEUS.

3. Does anyone know of somewhere that you can send Solos or ALS-8 formatted tapes to get listings and/or assemblies of them. I have ALS-8, but do not have access to a printer of any kind.

Sincerely,

Royce D. Bacon
8942 W. Lawrence Ave
Milwaukee, Wi 53225

Dear Tony:

Now that I have finally bitten the bullet and bought a printer, I can come out of my no-way-to-transmit-it shell and share some of my software with others. I have had my Sol-2Ø, Helios system for two years now and in this time have managed to fill approximately 45 disks with programs. I'll have to admit that some of my earlier programs are less than impressive.

Soon after I obtained my printer (Diablo 164Ø) I found that there was apparently no way to use the printer's keyboard while running Basic programs. Naturally, this presented an excellent oportunity to write another program. I am enclosing a listing of this program which I call SKIN. The name is an acronym for Serial Keyboard IN.

The console character input and character testing routines called from PTDOS are replaced with two routines which accept data from either the Sol's keyboard or the serial input port (printer's keyboard). A problem is created by the fact that to operate the Diablo in it's most efficient mode, occasionaly an ETX (ASCII Ø3) must be included in the output data stream. This is necessary to allow the printer to catch up with the data in the printer's buffer. When the printer encounters the ETX it sends an ACK (ASCII Ø6) back to the computer. SKIN checks for the incoming ACK and does not allow it to be echoed to the program using the PTDOS console input routines.

The output portion of the program can switch between the PTDOS console output routine and the printer. Which one of the outputs is used is governed by the byte stored in the Solos current pseudo output port location (ØC8Ø7H). If the byte is zero then the PTDOS console output routine will be used. If the byte is non-zero then output will be sent to the printer.

Another powerful feature of the Diablo 164Ø is the ability of the printer to recognize escape sequences. These escape sequences consist of the escape character (ASCII 1BH) and one or two more bytes depending on the desired function. These functions can include negative line feeds, vertical tabbing, incremental movement of the print head by 1/12Ø of an inch, etc. The printer driver routine (starting at label PTCON+3) includes the necessary logic to assure that once an escape sequence is started, an ETX will not be thrown in until the escape sequence is ended. (If an ETX is included in the escape sequence, interesting results can occur).

At the end of the program a short section of code is included which will allow a Basic program to output escape sequences to the printer through use of CALL statments . This section of code was necessary to implement a Basic program I wrote to do X-Y plotting on the printer. Several hidden features in PT's Diablo drivers will not allow them to be used for efficient X-Y plotting.

To use the enclosed I/O routines with EDBASIC it is sufficient to configure BASIC to use it's internal I/O routines. Apparently BASIC uses DOS CONIN and CONTST routines for all it's console input. Therefore, to use the printer keyboard, all that is necessary is to execute SKIN which will patch DOS's routines. The printer keyboard can then be used simultaneously with Sol's keyboard. To cause the output to be sent to the printer, it is necessary to SET OF=#1 and then POKE 512Ø7,1. To turn the printer off just POKE 512Ø7,Ø.

It should be noted that SKIN will work equally well with programs other than Basic programs. As a matter of a fact, I use it in place of the SETOUT command in PTDOS for program listings and the like. Note however, SKIN does not support the impressive bidirectional printing of PT's 161Ø driver.

Another very good use I have found for the simultaneous serial keyboard input portion of the program is in transfering programs from a friend's Sol-Northstar system to mine. The easiest way I have found to do this is to hook the two systems together through a serial cable. Then EDIT the file to be copied over. When he does a program listing out the serial port (as though he were doing a listing) it will be automatically entered into my editor. I normally restrict the baud rate to 12ØØ baud. (Laziness is the mother of invention).

CAUTION: For reasons unknown at the present time, I am not able to run SKIN and PT's 161Ø, Sol2 etc. drivers at the same time. There are some strange interractions.

By the way, if anyone is interested I have written a program in BASIC which will convert a NORTHSTAR BASIC program to PT Extended DISK BASIC. The Program will accept input from either a disk file or the keyboard and store the converted program on the disk. (Laziness again. Also opportunity to write another program).

Finally, I would like a schedule of advertising rates in PROTEUS. I have several programs which I would like to advertise for sale. Since mine is a programming for the love of programming operation, I plan to sell the programs for about the cost of reproduction. Also I am available for custom work for unreasonably low rates.

Keep up the good work. Proteus is an outstanding publication. For the record: I have worked with CPM and have found that compared with PTDOS, CPM is a dog.

Don L. Finley
236 Willow St.
Mt. Carmel, Tn.
36642

We used to charge for ads, but now they are free to member/subscribers, in the interest of spreading the knowledge.
– Stan

```
****************************************
**
*     SERIAL & KEYBOARD INPUT
*     SERIAL &/OR CONSOLE OUTPUT
*     DON L. FINLEY   2-22-80
*     NOTE: Due to code patching, following program will
*           not run in PROM
**
****************************************
      ORG ØC9ØØH
      XEQ START
*
*  START-UP INITILIZATTION
*  PATCH NEW ADDRESSES INTO PTDOS
*
START LXI SP,RAMTP
      LHLD SYSGLO   GET SYS GLOBAL POINTER
      LXI D,GLRCH   READ CHARACTER ROUTINE
      DAD D
      LXI D,CONIN   MOVE OUR NEW CONIN ROUTINE
      CALL MOVE     TO PTDOS POINTER LOCATION
      LHLD SYSGLO
      LXI D,GLWCH   POINT TO PT'S WRITE CHARACTER
      DAD D         ROUTINE POINTER
      PUSH H        SAVE THE ADDRESS FOR LATER
      MOV E,M       NOW, GET PT'S OUTPUT ROUTINE
      INX H
      MOV D,M
      XCHG          PUT IT IN H,L
      SHLD PTCON+1  AND PATCH IT IN OUR CONOUT ROUTINE
      POP H         NOW PUT ADDRESS OF OUR ROUTINE IN PT'S
      LXI D,A164Ø   POINT TO NEW CONOUT ROUTINE
      CALL MOVE     MOVE IT TO PT'S ROUTINE POINTER
      LHLD SYSGLO
      LXI D,GLTCH   POINT TO PT'S CONSOLE TEST FOR WAITING
```

Cont. pg 14.

Cont. from Pg. 13

```
      DAD D           CHARACTER ROUTINE
      LXI D,CONTS     PUT THE NEW ROUTINE IN PLACE OF IT
      CALL MOVE
      LHLD SYSGLO
      LXI D,GLBIO     POINT TO CONSOLE BINARY I/O FLAG
      DAD D           COMPUTE THE ADDRESS AND PATCH IT INT
      SHLD X4+1       OUR ROUTINE
      XRA A
      STA OFLAG       INITIALIZE TO PT'S CONOUT ROUTINE
      JMP  ØBCBØH     ALL DONE
*
* ROUTINE FOR MOVING ADDRESS IN D,E TO MEMORY
*
MOVE  MOV M,E
      INX H
      MOV M,D
      RET
*
* THE ROUTINE CONTS WILL BE USED IN THE PLACE OF
* THE PTDOS ROUTINE CONTST
*
CONTS IN KSTAT        TEST KEYBOARD STATUS
      CMA             ROLL IT OVER
      ANI Ø1
      RNZ             SOMETHING THERE IF <>Ø
      IN SSTAT        TEST SERIAL STATUS
      ANI 4ØH
      RZ              NOTHING THERE EITHER
      IN SDATA        SOMETHING THERE, BUT WHAT IS IT
      ANI Ø7FH
      CPI ACK         IS IT ACKNOWLEDGE?
      JZ X1           IF SO THEN CLEAR ACK FLAG
      STA CHARB       NOPE, SAVE IN CHARACTER BUFFER
      MVI A,1
      STA BFLAG       SET CHARACTER-IN-BUFFER FLAG
      ORA A           SET CHARACTER WAITING FLAG
      RET             GO HOME
X1    XRA A
      STA AFLAG       CLEAR ACK FLAG
      RET             RETURN AS THOUGH NO CHARACTER
*
*  THE ROUTINE CONIN IS TO APPEAR THE SAME AS PTDOS
*  ROUTINE CONIN
*
CONIN LDA BFLAG       TEST CHAR.-IN-BUFFER FLAG
      ORA A
      JZ X2           NO CHARACTER
      XRA A
      STA BFLAG       CLEAR CHR-IN-BUFFER FLAG
      LDA CHARB       GET CHAR. FROM BUFFER
      RET             TAKE IT BACK
X2    IN KSTAT        CHECK KBORD STATUS
      CMA
      ANI Ø1
      JNZ X3          SOMETHING ON KEYBOARD
      IN SSTAT        CHECK SERIAL STATUS
      ANI 4ØH         TEST IF RECEIVER FULL
      JZ X2           IF NOT, KEEP LOOKING
      IN SDATA        GET WHAT IT IS
      ANI 7FH
      CPI ACK         IS IT ACKNOWLEDGE?
      PUSH PSW
      JNZ X4          IF NOT, GO HOME WITH IT
      POP PSW         WAS ACK, GET IT BACK
      XRA A           AND CLEAR FLAG
      STA AFLAG
      JMP X2          RAN OUT OF DESCRIPTIVE LABELS
X3    IN KPORT        GET K'BORD INPUT
      PUSH PSW
X4    LDA GLBIO       CHECK IF BINARY FLAG
      ORA A           WANTS PARITY STRIPPED
      JZ X5
      POP PSW         NOPE
      RET
```

```
X5    POP PSW         YEP, STRIP PARITY
      ANI 7FH
      RET
*
*  164Ø DRIVER ROUTINE
*  NOTE: DOES NOT SUPPORT GAUDY BI-DIRECTIONAL PRINTING
*
A164Ø MOV B,A
      LDA OFLAG       SEE WHERE OUTPUT GOES
      ORA A           IF NON-ZERO THEN GOES TO PRINTER
      MOV A,B         IF ZERO THEN TO PTDOS
PTCON JZ Ø            OUTPUT TO CONSOLE, ADDRESS PATCHED
      LDA AFLAG       TEST IF NEED TO WAIT ON ACK
      ORA A
      CNZ WAIT
B164Ø IN SSTAT        GET SERIAL STATUS
      RAL
      JNC B164Ø       WAIT TILL XMITTER EMPTY
      LDA EFLAG       SEE IF ESCAPE SEQUENCE SET
      ORA A
      MOV A,B         GET CHARACTER BACK
      JNZ ESCE1       IF SO THEN SPECIAL HAND.
      CPI ESC         IS IT ESC.
      JNZ OUTIT       IF NOT, THEN OUTPUT IT
ESCSE OUT SDATA       OUTPUT ESCAPE
      MVI A,1
      STA EFLAG       SET ESCAPE SEQUENCE FLAG
      LDA NCHAR
      INR A
      STA NCHAR
      MOV A,B
      RET
*
* SPECIAL HANDLING FOR CHARS FOLLOWING ESCAPE
*
ESCE1 CPI Ø9
      JZ ESCE5        HT REQUIRES ANOTHER CHR.
      CPI ØBH
      JZ ESCE5        VT ALSO
      CPI ØCH
      JZ ESCE5        FF ALSO
      CPI ØDH
      JZ ESCE5        CR ALSO
      CPI 16H
      JZ ESCE5        SYN ALSO
      CPI 1EH
      JZ ESCE5        RS ALSO
      CPI 1FH
      JZ ESCE5        US ALSO
      OUT SDATA       SINGLE CHAR AFTER ESC
      XRA A
      STA EFLAG       CAN TURN OFF ESC FLAG.
      LDA NCHAR
      INR A
      STA NCHAR
      MOV A,B         GET CHAR BACK
      RET             GO BACK
ESCE5 OUT SDATA       ANOTHER ESC SEQU. CHAR COMING
      MOV A,B         LEAVE ESC. FLAG SET
      RET
OUTIT OUT SDATA       PRINT IT OUT
      LDA NCHAR       TEST IF NEED TO INSERT AN ETX
      INR A
      STA NCHAR
      CPI 1Ø1         PUT IN AFTER 1ØØ CHARACTERS
      JC D164Ø        IF LESS THAN
      MVI A,1
      STA AFLAG       SET ACK FLAG
      STA NCHAR       ALSO CLEAR CHARACTER COUNT
C164Ø IN SSTAT        OUTPUT ETX CHARACTER
      RAL
      JNC C164Ø
      MVI A,ETX
      OUT SDATA
```

Cont. Pg. 15.

```
D164Ø MOV A,B       DONE FINALLY
      RET
WAIT  IN SSTAT      GET SERIAL STATUS
      ANI 4ØH
      JZ WAIT       WAIT FOR SOMETHING
      IN SDATA      SOMETHING THERE
      ANI Ø7FH      WHAT IS IT?
      CPI ACK       IF ACK, THEN CARRY ON
      JZ WAIT1
      STA CHARB     IF NOT, SAVE CHAR IN BUFFER
      MVI A,1
      STA BFLAG     SET FLAG
      JMP WAIT      KEEP WAITING
WAIT1 XRA A         CLEAR WAIT-FOR-ACK FLAG
      STA AFLAG
      RET
*----------
*  EQUATES
*----------
CR EQU ØDH   CARRIAGE RETURN
LF EQU ØAH   LINE FEED
TAB EQU Ø9H  HORIZONTAL TAB
FORM  EQU ØCH   FORM FEED
ESC EQU 1BH     ESCAPE
ETX EQU Ø3H     END OF TEXT (SENT BY SKIN)
ACK EQU Ø6H     ACKNOWLEDGE (SENT BY PRINTER)
KSTAT EQU ØFAH      KEYBOARD STATUS PORT
SSTAT EQU ØF8H      SERIAL STATUS PORT
KPORT EQU ØFCH      KEYBOARD INPUT PORT
SDATA EQU ØF9H      SERIAL DATA PORT
CONOA EQU ØBC9FH    PTDOS WRITE CHARACTER TO CONSOLE
SYSGLO EQU ØBCA5H   PTDOS SYS GLOBAL POINTER
GLRCH EQU 15H       CONSOLE READ CHARACTER ROUTINE POINTER
GLWCH EQU GLRCH+2   CONSOLE WRITE CHARACTER ROUTINE POINTEF
GLTCH EQU GLWCH+2   CONSOLE TEST FOR WAITING CHAR. POINTER
GLBIO EQU 3CH       CONSOLE BINARY I/O FLAG
*
*  STORAGE
*
EFLAG DB Ø   ESCAPE SEQUENCE FLAG
BFLAG DB Ø   CHAR-IN-BUFFER FLAG
AFLAG DB Ø   WAIT FOR ACK FLAG
NCHAR DB Ø   NUMBER OF CHARACTERS OUTPUT
CONAD DW Ø   ADDRESS OF PTDOS OUTPUT ROUTINE
OFLAG EQU ØC8Ø7H    USE SOLOS OUTPUT PORT LOCATION
CHARB DB Ø   CHARACTER HOLDING BUFFER
      DS 2ØH
RAMTP EQU $  SAVE ROOM FOR STACK FOR INIT. ROUTINE
      ORG ØCBØØH
*
* PLOT DRIVER FOR BASIC
* FOLLOWING ROUTINE IS CALLable FROM BASIC PROGRAMS TO
* USE THE DIABLO 164Ø PRINTER AS AN X-Y PLOTTER
* THE PROPER CALL WOULD BE Z=CALL(51968,C)
* WHERE: Z IS A DUMMY VARIABLE
*        C IS THE ASCII CODE FOR CHAR. TO BE OUTPUT
*
      MOV A,E
      JMP ØBC9FH    PTDOS CONOUT
      END
```

Dear Stan

I'm glad that I found out about your Newsletter, especially since PTC folded up.

All those Change Notices and Updates I ordered from PROTEUS really fixed us up. Also, the Disk Memory System Manual was a real find; we didn't have a Theory Of Operation section. We received our Helios II in Sept. 1977 and had only received notices up thru #4.

My brother and I jointly own a computer system consisting of the following:

IMSAI mainframe
TDL Z-80 processor
PTC VDM-1, CUTS board, and Helios II
Computalker
Speechlab
Heathkit H14 printer
Addmaster paper tape reader
Software Tech. Music System
Objective Design Programmable Character Generator
Cromemco Bytesaver
MITS 88-4 Parallel I/O board
IMSAI SIO 2-2 serial I/O board
Around 48k of static ram

I wrote our operating system from the PTC CUTER and Technical Design Labs 2K ZAPPLE monitors. I used the command buffer and cassette tape features from CUTER, and the extensive command set from the ZAPPLE monitor (it has 26 commands) along with the entry vectors for both monitors. It occupies a little over 4k, and the assmebler source listing is 72 pages long.

I have written some device drivers for the Heathkit H14 printer; namely, two that just print as the data is received (one at 80 characters/line and the other at 132 characters/line) and another that formats the data 55 lines/page, at 80 characters/line with the left margin indented 3 spaces so that the edge may be punched. I use the formatted driver for BASIC listings. If you are interested, I can send PROTEUS source listings of them later.

We noted that some of the owners of Helios disk systems were disappointed that the hardware write-protect option was not included with the PerSci drives. Well, we added the protect feature to our drive using a toggle switch and an LED indicator. See figure 1 for the schematic. P20 is along the top edge of the right-hand pcb (data and interface board) on the disk drive. When the LED is on, unit 0 is write protected. NOTE: some programs and commands (such as DO) require that this switch be off (write enabled) because of housekeeping that is done on the system disk. If you get a FILE ID CONFLICT or BLOCK SIZE CONFLICT error, it may be due to having the switch in write-protect when an operation tried to write on the unit 0 disk. While the PTDOS write-protect attribute is OK, it doesn't help when the system runs amuck and tries to write all over the system disk (which has happened to us, twice).

We are using an Artec 32K memory board (with 16k of ram on it) for PTDOS. We've had no DMA problems with this board, but it runs very hot. We're planning to replace it with a lower power board in the future. With almost a full computer (the IMSAI holds 22 boards), a lot of heat is generated as is.

Cont. pg. 16.

Cont. from Pg. 15.

By the way, do you or anybody else know what the SOLGO file on the FORTRAN disk is all about?

Keep up the good work on the Newsletter. We will write again.

Anyone who has any questions about operating Helios with a mongrel system like ours, or has anything they want to tell us may reach us thru this Newsletter.

Sincerely yours

allen

Suffolk, VA 23435

Note: I have the same Artec board but with the full 32K and a similar 32K board with the same memory chips —they both run hot. BUT, I lowered the backplane "+8v" to 7.5v with a bucking transformer, and I can run both together in my Sol without heat problems. Joe Maguire wrote an article in Proteus News a while ago explaining one way to do this.

—Stan

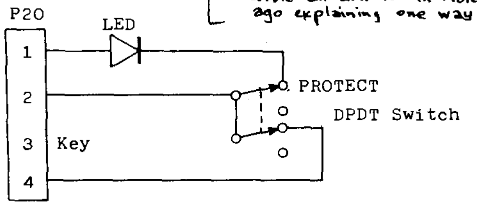


Figure 1

COMBINED TAPE AND DISC STORAGE.

Dear Sol Brother Tony,

I was startled to see that you had printed some comments I had scribbled on the questionaire attached to the subscription renewal form, and I would like to clarify them.

Let me start by reiterating my suggestion that the practice of using combined tape and disc storage on mainframe machines seems to be equally valid for micros. Admittedly audio cassette recorders are slow, but they are very reliable and the media is cheap.There are undoubtedly applications where one would be content go away while the machine is plodding along doing some mundane task such as reading and printing a mailing list.

However there are faster tape storage systems such as the MECA Alpha-1 which offer a lot of storage in relative dollar terms. Incidentally it would be interesting to hear of anyones experience with an Alpha-1 and Sol. I was prepared to buy one of these units but MECA required a large order before they would adapt ProTec ECBasic to their system and I did not wish to switch to MICROSOFT at the time.

I have repeatedly postponed buying a disc system because of the rapid changes that have taken us from single sided single density to multi megabyte hard discs, along with changes in format and executive software, in three years. I keep waiting with my ear to the ground to hear if the latest hardware developement is proving reliable.

My current favourite would be a single sided double density NORTHSTAR plus a MECA Alpha-1.

To take speculation several steps further my ideal home micro system would include the hardware just listed plus a real time clock/vectored interupt board and a software System Executive that would timeshare foreground/background programs according to priorities requested through the software.

With such a system home process control and monitoring could proceed in the foreground while data processing and games could run as background programs. Although the control and monitoring functions would have the highest priority and would consequently interupt other activities it is doubtfull that they would consume more than ten percent of the CPU time.

What I have described is a duplication of the now obsolete IBM 1800 data aquisition and control system. However I don't think the system design is obsolete because of its versatility. There is no doubt in my mind that versatility and reliability are the prime requisites for a home computer, and that is the kind of system I intend to have in my underground house when I finish building it.

B.W.(BERNIE)LITTLEJOHN
CANADA V2G 2P1

THE MECA BETA-1 TAPE SYSTEM AND THE SOL

The Beta-1 Tape System, available from MECA (7026 O.W.S. Road, Yucca Valley, CA 92284) is a digital tape mass storage system that is suitable for use with many microcomputers, including the SOL. The standard unit, which comes complete with controller, power supply, Phi-Deck, enclosure, cable and manual, sells for $399, with additional drives (up to four drives total per controller) priced at $270 for driver, enclosure, and cable with connector. Double density, double speed, and serial (RS232) interface options are supposed to be available at higher cost. In this article, I will only discuss the standard unit, since that is what I have.
The Beta-1 records two tracks of information on a certified digital cassette. Data is written in blocks of 256 bytes, and a 300 ft. tape will hold 1000 blocks on each track, for a total of 512,000 bytes storage per tape. The parallel interface transfers 500 bytes per second on the standard unit, which is over four times the speed of SOL's CUTS audio tape interface. Also, the Beta-1 can find any block on the tape without having to read the preceding blocks (although it obviously does have to skip over them!) with a high-speed search rate of 100 inches per second, or twenty times the read/write rate. On a 300 ft. tape, this translates to a worst-case access time of 36 seconds. All in all, while it doesn't compare to a floppy disk in speed, the price isn't bad, and the capacity is reasonable.
Now comes the interesting part--interfacing the Beta-1 to the SOL. While MECA currently offers TRS-80 and Apple packages with the Beta-1, no such software/hardware package is available for the SOL. This is not as bad as it may seem, for two reasons: the SOL's parallel port and the Beta-1 parallel interface are reasonably compatible, and the Feta-1 controller contains an 8035 microprocessor with a 1K byte EPROM program that accepts quite a few ASCII-encoded commands. Of course, it's not all that simple, either. The SOL has its DB-25P connector, while the Beta-1 cable has a total of 34 lines. What I hope to describe in this article is the way I connected these two, along with some of my observations on using the Beta-1 with a SOL.

Cont. Pg. 17.

Cont from Pg. 16.

The diagram on the next page (Figure 1) shows the connections to be made between Beta-1 lines and SOL parallel port lines. In order to make the connections, you need to remove the connector from the Beta-1 ribbon cable, and you need to get a 25-conductor cable with a DB-25S connector at one end, for the SOL side. You must then separate the individual wires in each cable, and strip the last inch or so for the connection.

Now, I am a typical software hacker, and I avoid most fiddling with any kind of hardware as much as possible (mainly out of fear of really screwing things up!) In this case, however, I had an accomplice (I talked my father into helping me), so I plunged right into the work. After several false starts that ended less than successfully, I finally figured out how to attack the problem. (Along the way, I had to learn how to carefully separate and strip those 59 individual wires!) The big problem is that directly connecting wire to wire quickly yields a bundle of wires that you can't make any sense of, and chnces are you will end up connecting the wrong lines along the way. I am careful, at least in checking my work for mistakes, and I discovered quite a few the first times I tried this. Then I hit upon the idea of connecting the cable wires in sequence to a line of short wires, with the short wires pre-arranged in the proper order for proper connection between the two cables. I used a perfboard to hold the crossed wires, with the two lines of bare wire ends forming reasonable places to connect the Beta-1 and SOL cables. I then twisted together the corresponding wires, and finally checked for proper connection by testing from the Beta-1 connector end to the corresponding pin in the SOL connector with an ohmmeter. Once I assured myself that the wires were correctly connected, I reconnected the Beta-1 end and plugged the other end into the SOL, turned both units on, put the SOL in terminal mode with port two selected for I/O, and saw the Beta-1 respond with its "ready" sequence, BELL-CR-LF! (Actually, I had some problems in between that I haven't mentioned because they aren't relevant to the interfacing.)

To summarize in full what I did:

(1) Removed the Beta-1 cable's "computer" end connector, and separated and stripped the individual wires.
(2) Obtained a 25-conductor cable to the SOL parallel port, and separated and stripped the wires at one end also.
(3) Built a perfboard with two "connection strips" consisting of the ends of short telephone-type wires, which were inserted crossed in the appropriate order, so the cable wires could be connected in sequence. (A pair of terminal strips with wires cross-connecting them, would be even better, but I didn't have any available at the time.)
(4) Connected the cable wires to the "connection strip" wires.
(5) Re-attached the Beta-1 connector and connected the SOL end.

There are some additional steps involved:

(1) The SOL's "data acknowledged" input control line requires the corresponding Beta-1 signal to be inverted. I did this by using a 7406 IC on the perfboard, connecting the Beta-1 signal to pin 1, the SOL line to pin 2, and the +5V and ground lines to the Beta-1 lines of the same function. I added a 470-ohm pullup resistor on the SOL output on the advice of a friend (the resistor is connected from the +5V line to pin 2 on the 7406, with the SOL line coming off pin 2 also.)
(2) In order for the Beta-1 to supply the +5V needed by (1), you have to install jumper W30 on the Beta-1 board, following the disassembly/assembly instructions supplied with the unit.
(3) Three Beta-1 lines and four SOL lines are left unconnected, and the ten Beta-1 ground lines and the SOL's single ground are connected together.

Now for some observations:

Using the Beta-1 with the SOL means having to write driver routines. I considered writing replacements for the CUTS routines in SOLOS, but that seems to be too much work for starters, so I'm writing routines that will allow ALS8 to manipulate Beta-1 files, and allow the ALS8 assembler to directly assemble from a Beta-1 tape file. The Beta-1 controller program accepts commands like: open file, kill file, list directory, read, seek, write, and rewind. The driver program is fairly straightforward, but you do have to know what you're doing to write it. Eventually, I expect to have patches for the available SOL software to use the Beta-1; however, that seems to require some operating system level program design.

The Beta-1 system will probably be much more useful with at least two drives. The way the controller works, drive 0 has tracks 0 and 1, drive 1 tracks 2 and 3, etc.; thus, the software has little need to concern itself with number of drives. However, for backup copies and tape-to-tape processing of data, two drives would be nice.

The fact that file names are restricted to five characters, and files cannot be more than 256 blocks long, are not restrictions that you couldn't live with, although emulation of CUTS files is made more complex. I feel that a separate operating system would be the best answer to the problem, but I am not writing one just yet.

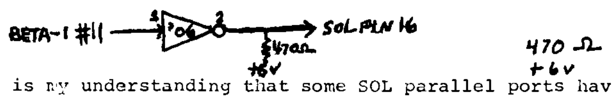
In summary: the Beta-1 provides mass storage capacity comparable to a single-sided double density 8" floppy, at a lower price but also at lower speed; it is an improvement of considerable magnitude over audio recording. The fact that no S-100 board is needed to interface the Beta-1 makes it attractive for the SOL, especially since the parallel interfaces are reasonable to connect. Software is a problem, but if enough people acquire Beta-1's, software should become available.

Above all, if you get a Beta-1 and try to duplicate my efforts, be sure to proceed carefully and double-check your work before you power up.

Figure 1

Beta-1 Signal #	Direction	SOL Parallel Port Pin #	and Signal (Beta-1 name)
1	==>	Note A	+5V
2	<==	25	Input data 0
4	<==	24	Input data 1
6	<==	23	Input data 2
7	<==	17	Input strobe
8	<==	22	Input data 3
10	<==	21	Input data 4
11	==>	Note B; 16	Data acknowledged
12	<==	20	Input data 5
14	<==	19	Input data 6
16	<==	18	Input data 7
20	==>	6	Output data 0
22	==>	7	Output data 1
24	==>	8	Output data 2
25	<==	5	Data acknowledged
26	==>	9	Output data 3
28	==>	10	Output data 4
29	==>	4	Output data ready
30	==>	11	Output data 5
32	==>	12	Output data 6
34	==>	13	Output data 7
3,5,9,13,15, 17,19,23,27,31	<=>	2	Ground
18,21,33		1,3,14,15	(Unconnected)

Notes:
(A) With Beta-1 jumper W30 installed; supplies +5V for (B)
(B) Beta-1 signal 11 is inverted by circuit below before going to SOL pin 16



(C) It is my understanding that some SOL parallel ports have their pins in different positions from others, depending upon the revision level. Mine is a Rev. E SOL. Check yours before starting!

Note: The portable PTDOS I talk about on page 8 could be interfaced to a block-addressable tape like the Meca. Wouldn't that be something?! Performance would be slow, but the system would be slick. When you finally add a disk, it would integrate into the same system. Stan

Want to correspond with owners of Houston Instrument Hi-Plot plotter for purpose of exchanging information/programs. William Graham, 3512 Garden Drive South, New Port Richey, Fl 33552

We are in the process of attempting to identify computer software materials for the instruction of elementary and middle school math skills. We would like to correspond with those who could tell us what material they have found effective and thus would recommend it to us. Paul E. Peknik, Jr. Stamford Public Schools, 195 Hillandale Avenue, Stamford, Conn. 06902

WANTED (Working or not) Micropolis Disk Controller Board #1071 and Micropolis Diagnostic software disk. A.A.Bonetti, PO Box 294 Petaluma, Calif. 94952. (707) 762-9514

FOR SALE:
Proc.Tech. Equipment..all items have manuals.
1. 32KRA-1 Memory Board with all parts to upgrade to 64K(not including RAMs) NEW! $325.00
2. SolPrinter2 metal wheel Diablo printer, NEW $1600.00
3. SolPrinter2-E plastic wheel Diablo printer, slightly used $1500.00
4. Sol System III-A with 64K memory in excellent condition with WordWizard, MailMaster, BASIC and PTDOS software, very slightly used. $2900.00

Analog Digital Research Co.
408 13th St. Suite 545
Oakland, Ca 94612
415-462-3394

COD orders accepted with 25% deposit. VISA and MasterCharge accepted. Items 2, 3 and 4 will be shipped freight collect, on 32K boards we pay postage and insurance. In California please include 6% (or in BART areas, 6.5%) sales tax.

SOL / HELIOS PARTS FOR SALE

I have been contact with some people who bought quite a lot of Processor Tech. equipment. They have agreed to sell them through me at what I consider a very resonable price. There is a good supply, but I don't know how long they are going to be willing to sit on their investment, so, if you may want to buy some spare parts now before they get harder to find. (And they will get harder to find!)

Send your orders (cash in advance, please) to:
Tony Severa's Data Service
131 Highland Ave
Vacaville, Ca 95688 (Expect 4-6 weeks FOR DELIVERY.)

Complete System III (Includes Sol-20, 48K memory, Monitor, Helios Disk system and Centronics 700 printer).....$4000.00
(Or will sell individual items to highest bidder.)

Fully wired and operational Sol Mother boards.......$350.00
Fully wired and operational 32NKRA ....can be expanded to 64K..$350.00
Fully wired and operational keyboards...............$130.00
ALS-8 boards (includes PROMS)......................Highest bid
Whole Subsystem 'B' in IMSAI frame.................$1000.00 or highest bidder.

103001.....Rev B Backplane boards.....................$7.50
107001.....2708 Personality modules...................$7.50
110211.....Regulator..................................$7.50
110221.....Rev C Backplane boards.....................$7.50
110201.....Rev C Sol main logic board................$50.00
301004.....Rev C Helios Formatter board...............$7.50
301004.....Rev D Helios Formatter board...............$7.50
302001.....Helios Regulator board.....................$5.00

Sol Board Set (107001,110211,110221 and 110201 listed above)
Documentation not included..........................$65.00

G-2 Software (on cassette)
Sol-20 Microsoft Extended Basic (16K)...............$25.00
Beat the House (Requires Microsoft Basic).....$7.50
Clinic (Requires Microsoft Basic).............$7.50

The following list of people/companies continue to support the Sol/Helios system. We cannot guarantee that they provide a quality service. If we hear any negative reports regarding anyone who has been listed in our newsletter and the negative reports are more than just a miss-understanding and occur more than once, then we will notify you that they have been taken off of our list.

Tony's Data Service
131 Highland Ave
Vacaville, Ca. 95688
707-448-9055
(I have several people whom I can get to repair your system. I will check your system out before shipping back to you.)

Ray Pohl
3388 Moraga Street
Lafayette, Ca 94549
415-283-6630
(Can also be utilized by Tony's Data Service)

Albert Bonetti
PO Box 294
Petaluma, Ca 94952
707-762-9514
(Can also be utilized by Tony's Data Service)

Computer Mart of New Jersey, Inc.
501 Route 27
Iselin, N.J. 08830
201-283-0600

Mr. Leland Mull
504 Kibler Circle
Vienna, Virginia 22180
703-938-8219

Micro Automation Repair Service
P.O. Box 328
Binghamton, N.Y.
607-724-5115

Basic Computer Store
Anarbor, Mich

Basic Computer Store
Akron, Ohio

Lee Felsenstein (Golemics, Inc.) wants to gather suggestions for appropriate additions to and versions of the personality module. Lee has already mentioned his commitment to provide the VDM-2 (see Dr. Dobb's Journal, Nov./Dec. 1979) in a version for the Sol, which would require a replacement personality module containing the new screen driver.

In addition to this, Lee is working with some other people who have relocated the Solos module to page "F", and Lee has provided a hardware modification which relocates the entire on-board RAM/ROM/ VDM to that page. Other possibilities include "phantoming" out the personality module under control of a port, arranging for the capability to replace the module without crashing the program, and perhaps some kind of ROM bank switching to extend the PM's effective size.

Lee had announced his intention to attend the Dec. 4th PROTEUS meeting but was detained by urgent business. Therefore he requests that anyone having suggestions on this topic write to him at Golemics, 1407 Addison St., Berkeley CA 94702. Lee will of course be available to discuss this at the next meeting of the Homebrew Computer Club,

---

If you had not heard, the North Star Winchester plugs into the Parallel port on the Horizon. By wiring an adapter cable, it will be possible to attach the drive directly to a Sol through its parallel port on the back. This would allow a Helios II to access 18-19 Meg of storage. The Helios would make an excellent back-up unit for the data on the hard disk. We would like to hear from other Helios owners about configuration preferances, ie should the North Star be the first drives in the system (0-4), with the Helios II as high number backups, or should the hard disk be drive 0-3d, or whatever? We would appreciate any and all comments about this! We could set the system up to boot off the hard disk, but this would require a new personality module.

Bill Blomgren
Microcompauter Resources, Inc
3000 Medical Park Drive, Suite 108
Tampa, Fl. 33612

Subject: CP/M2 for SOL on North Star DD

Attn: Tony Severa,

I am something of an operating system freak and I thought that members may be interested in a few comments on CP/M2 from Lifeboat Associates. The byte at 03H is known as the IOBYTE and controls access to physical devices. The following is the gist of a letter I sent to Lifeboat today after they were unable to give the info over the phone.

The user area in CP/M2 is 100 Hex bytes below the location it held in CP/M Ver.1.4 . For SOL users this means that the byte at 5B1D in Ver 1.4 or at 5A1D in CP/M2 in the 24K distribution version (BB1D or BA1D in my 48K version) is placed at byte 3H when a cold boot is done and therefore this byte controls the IO ports. Lifeboat's standard version for SOL expects that the parallel port will be the user's printer port whereas in my system, and I expect most others, the serial port is the printer port in use.

I therfore insert 64H at the above location, do a warm boot and then do the SAVEUSER routine. Each time the system is cold booted after this, byte 3H is 64 and the serial port is the list port.

Any Sol user who has Lifeboat CP/M ver 3 for HELIOS will find that this same thing can be done by changing the byte at BC18H in the 48K version.

For any SOL owner who has access to a 24 X 80 terminal and wishes to write CP/M software for general distribution the IOBYTE can be set to 27H and the terminal connected as a serial in and out device. The terminal will then be using the SOL as a host computer and full use of the bigger field of the terminal can be had. I am installing an ALPHA-MICRO system at the present time and I found the use of CP/M in this fashion good training in getting used to the quite differant style of operation after being used to PTDOS.

I would also like to comment on TAD Enterprises' CP/M adaption for PT's Extended Cassette Basic. Anyone with a SOL and CP/M should get this terriffic package.

J.G.Zeratsky

Dear Tony,

1. I just got an IDS-440 "PAPER TIGER" printer for my Sol and I need some help in writing a driver for it. I am inexperienced when it comes to assembly language programming. What I do now is set my serial plug as output paort then everything goes to the printer not ver effective in some cases. Also which port is better to hook a printer up to - the serial or the parallel port?

2. I will soon be getting a VISTA-200 with 2 drives, would like to know of anyone else who has this combination. After I get all the bugs out and am at full potential I would be glad to volunteer to convert some of the cassette based software to VISTA format. For you information it is a double density drive, 10 hard sectors, 512 bytes/sector and 40 traks for a capacity of 204800 bytes.

Bob Freeman
1057 Ouray Ave
Grand Junction, Co 81501

TABLE OF CONTENTS

ANNOUNCEMENTS

PROTEUS / NEWS (FORMERLY CALLED SOLUS)

Publisher: PROTEUS, 169Ø Woodside Road, Suite 219,
Redwood City, California 94Ø61, U.S.A.
Editor: Tony Severa, 131 Highland Ave, Vacaville
California, 95688, U.S.A.
Subscriptions: $18 per calendar year in U.S.A., $18 in Canada, $24 elsewhare (U.S. dollars only). All subscriptions expire at the end of the calendar year (Dec. 31). Mid-year subscribers will receive back issues for current year. Back issues for prior years are sold in annual sets, while available.

Instructions to contributers: Letters and articles may be submitted in camera-ready form or on Sol/Cuts cassettes or Helios (PTDOS) diskettes. Camera-ready copy should be single-spaced, in a single column of 6 1/2 inch width, and with clean, dark type. Corrections can be made invisibly with opaque correction fluid ("liquid paper"). Please use a new ribbon. Machine readable articles should be compatible with Solos, Cuter, PTDOS input routines. Mdia will be returned only if requested.

PROTEUS
169Ø WOODSIDE ROAD, SUITE 219
REDWOOD CITY, CALIFORNIA 94Ø61
U.S.A.

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

18104

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

May/June 1980
Vol. 3, # 3

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

Single Issue....$3 (US)

EDITOR'S COLUMN

by Stanley M. Sokolow

It is with great reluctance and some regret that I have decided to resume the editorship of Proteus News. Tony Severa actually has prepared this issue and the one to follow, but I have held them while I decided what to do.

Tony has done a satisfactory job as editor during the period he held the post, but I have become increasingly dissatisfied with some of his other activities in connection with Proteus. I don't know if this was due to taking on more than he could handle (he was involved with Apple software, and more recently with a NorthStar User's group announced on page 14 of this issue), or if it was just due to oversights, but in any case he was not as responsive as I felt he should be. Consequently, I felt obligated to take over again. If any of you are still waiting for something Tony was supposed to send, please let me know. This issue was completed before I made the decision to retake the editorship, so I couldn't change all of the content. Therefore:

> Please disregard any references inside this issue to Tony Severa's address for the Helios library, document updates, newsletter articles, etc. Contact me at Proteus address, instead. For anything other than Proteus activities, such as his International NorthStar Users Group, continue to contact Tony as you see fit.

All articles, letters, etc., for publication should now go to the Proteus address in Redwood City, shown above. If you have submitted any articles recently that have not been published, please send me another copy of the story if possible. Likewise, please send Helios library contributions and orders to Proteus until I announce a new librarian.

I HAVE ALMOST ALL OF THE SOURCE CODE!
See story on page 17.

WEST COAST COMPUTER FAIRE AGAIN

Once again, Proteus has the opportunity to have a booth and/or meeting at the Computer Faire in San Francisco. It will take place the first week in April, 1981, so lets all plan to get together there. I think it's going to be April 3, 4, 5 in Brooks Hall. If anyone would like to exhibit something nice he has done with his system, please let me know.

CONTINUED ON PAGE 20

DISCOVER

UNDERSTANDING AND USING YOUR SOL I/O

Dear PROTEUS PEOPLE,

Here I am again with more goodies for all my SOL brothers. This time I am sending along some notes that I have drawn up on **UNDERSTANDING AND USING YOUR SOL: INPUT ROUTINES.** I have begun a rather ambitious project that involves writing documentation for the SOL. Now I know that Processor Tech provided pretty decent documentation, but what I am trying to do is gather into one place information that is often scattered to the four winds. I plan to do things by topic, and be very redundant. By repetition, maybe some of this information will finally make more sense to some of us. I think that each of us can use a little help in trying to use our SOL to the fullest.

I have some requirements I want to set down. Firstly, I want this stuff placed in a logical, CONSECUTIVE manner within the hallowed pages of our beloved PROTEUS, not "continued five pages from here". To be of maximum utility, it should be presented as a WHOLE, not scattered all over the place. Secondly, I want this letter included with the text, because I have something to say to our membership: To make this particular endeavor work is going to require input from **YOU**, the reader. Surely you will notice a slight error, or perhaps even a glaring omission on my part as you read over these pages. Perhaps you have some thoughts on how it could all be better organized. PLEASE send me your comments, suggestions, and yes, even your well-earned criticism. I will then do my best to incorporate all of the feedback into a more coherent manuscript. I plan to cover a lot of various topics, including OUTPUT ROUTINES, TAPE ROUTINES, VIDEO MANIPULATION, SCANNING OF PARAMETERS, and in general digging into all those cute subroutines in SOLOS, and detailing such things as entry points, parameters passed, uses, etc.. I realize that this is, to a certain extent, a duplication of information that is already available to anybody who has the time and patience to forage through the assembly listings and various manuals. The difference is that THIS information will be organized by topic, thereby maximizing the chance that you will be able to find what you are looking for. As I get various topics beyond a certain stage, I will be presenting them for publication in PROTEUS. Once the final manuscript is available, I plan to print the thing as a complete set. There are still details to be worked out, but I plan to do a quality job at a rock bottom price. In the meantime, the sections that appear in PROTEUS will serve, hopefully, as a sort of stimulus for you and I to continue this out to the bitter end. **IT WILL NOT GET DONE WITHOUT EACH OF US DOING OUR BIT, HOWEVER SMALL.** When writing, if you desire a response, PLEASE enclose a self-addressed, stamped envelope.

CONTINUED ON PAGE 2

UNDERSTANDING AND USING YOUR SOL...CONTINUED FROM PAGE 1.

*** GENERAL OVERVIEW OF INPUT ROUTINES ***

SINP:	**CØ1F:**	**Uses pseudoport specified by IPORT (C8Ø6).**	
AINP:	**CØ22:**	**Uses pseudoport specified by "A" register.**	
KSTAT:	**CØ2E:**	**SOL KEYBOARD.**	**(Pseudoport Ø).**
SSTAT:	**CØ42:**	**SOL SERIAL PORT.**	**(Pseudoport 1).**
PASTAT:	**C2DD:**	**SOL PARALLEL PORT.**	**(Pseudoport 2).**
ERRIT:	**C2CB:**	**USER-DEFINED INPUT "DEVICE".** **(Gets address from UIPRT at C8ØØ-C8Ø1).**	**(Pseudoport 3).**

CHARACTERISTICS COMMON TO ALL INPUT ROUTINES

CHARACTER RETURNED IN "A": "Z" FLAG SET IF NO CHARACTER FOUND: ONLY THE "A" REGISTER AFFECTED. NO STRIPPING IS DONE OF THE MSB (Most Significant Bit). STATUS IS DESTROYED.

```
TYPICAL CALLING PROGRAM:
PROG    CALL INPUT      ;check for a character.
        JZ PROG         ;If none available, wait.
        ANI 7FH         ;Strip off MSB.
        *** stripped character is now in "A" ***
```

Sometimes you will not want to strip off the MSB, such as when checking for special SOL keyboard characters, but in general stripping should be done, especially when using input devices other than the keyboard. Many a programmer has forgotten to strip off the MSB and had it come back to haunt him later.

There are also times when you just want to check to see if anything at all was entered, but you can't afford to get hung up in a loop. For these cases, a temporary register in memory may have to be used to hold the data for possible future use, since it will otherwise not be recoverable because the first time the character was read, the status information was reset. The following routine is an example of temporary storage.

```
CHECK   CALL INPUT      ;Check for a character.
        STA TEMP        ;store it in TEMPorary location.
        *** a ØØ code is stored if no character ***
        *** "Z" flag is set if no character ***
        *** if there WAS a character, it is in TEMP ***
```

A program that can recover a stored character, or get its own if none is stored is given as an example.

```
GETIT   LDA TEMP        ;check TEMPorary storage first.
        ORA A           ;If empty, set "Z" flag.
        JNZ GOTIT       ;If OK, skip loop.
LOOP    CALL INPUT      ;if nothing, try again.
        JZ LOOP         ; keep trying.
GOTIT   ANI 7FH         ;Strip off MSB.
        PUSH PSW        ;Save character.
        XRA A           ; "A" has a ØØ now.
        STA TEMP        ;"RESET" TEMP so it is "empty".
        POP PSW         ;Recover character.
        *** character is now in "A" ***
        *** TEMP contains ØØ ***
```

*** THE MAJOR ROUTINES ***

THE SINP ROUTINE

SINP: CØ1F: Pseudoport number recovered from IPORT (C8Ø6): Character returned in "A": "Z" flag set if no character found: Only the "A" register affected. No stripping is done of the MSB (Most Significant Bit). If a character was found, status is reset. Default device is keyboard. RESET and certain errors will cause the default condition.

SINP is the SYSTEMS INPUT routine. This is the prime entry point for normal systems input. It is accessed by a CALL CØ1F, and returns the character in the "A" register. If no character was ready, then the "Z" flag is set upon return. The routine preserves all registers except "A", in which the character is returned. If a character was found, then status is reset.

SINP accesses one of four possible input devices, which will be dealt with in detail later. For now, it is enough to say that Ø=the Keyboard, 1=the Serial Port, 2=the Parallel Port, and 3=a User-Defined Input Routine. Which of these pseudoports will be used depends upon the contents of a status byte called **IPORT**, located at **C8Ø6**. The binary value from Ø-3 found at this location will be loaded into the "A" register, and then control is passed to the AOUT routine (which is discussed next). The default value at IPORT is Ø, for the SOL keyboard.

The value at IPORT may be changed in two ways. From the keyboard, when in the SOL COMMAND MODE, typing SE I=X (where X is a value from Ø-3), will cause X to be the current pseudoport. It should be noted that a RESET will cause the default value of Ø to be loaded again.

The second way to load a value into IPORT is under program control. For instance, the series of instructions MVI A,3 / STA C8Ø6 would cause the User-Defined Input Routine to become current. By the same token, a program can find out what the current pseudoport is by using the instruction LDA C8Ø6.

THE AINP ROUTINE

AINP: CØ22: Enter with pseudoport # in "A": Character returned in "A": "Z" flag set if no character found: Only the "A" register affected. No stripping is done of the MSB (Most Significant Bit). If a character was found, status is reset.

INP stands for "A-INPUT". The AINP routine is accessed by placing a pseudoport code in the "A" register, and calling CØ22. The pseudoport code can have the following values:

Ø = KSTAT: CØ2E, the SOL KEYBOARD
1 = SSTAT: CØ42, the SOL SERIAL PORT
2 = PASTAT: C2DD, the SOL PARALLEL port
3 = ERRIT: C2CB, the SOL USER-DEFINED ROUTINE
(It vectors output to the address found at UIPRT)

When the AINP routine is called, the value in the "A" register is used to vector to one of the above mentioned pseudoport routines. It is up to the user's program to insure that the "A" register contains a pseudoport number from Ø-3. (The AINP program will discard all but the two least significant bits, as it contains its own ANI 3 instruction).

*** PSEUODOPORTS ***

BESIDES USING SINP AND AINP, PSEUDOPORTS MAY BE DIRECTLY CALLED

Ø: KEYBOARD: KSTAT: CØ2E

Character returned in "A": "Z" flag set if no character found: Only the "A" register affected. No stripping is done of the MSB (Most Significant Bit). If a character was found, status is reset.

The Keyboard is pseudoport Ø, the default pseudoport. Besides being available via calls to SINP and AINP, it may be directly accessed via a CALL KSTAT (CALL CØ2E). It is suggested that either SINP or AINP (with "A" set to Ø), be used to access this routine, as these are standard entry points for all SOL systems, but the address of KSTAT may vary.

CONTINUED ON PAGE 3.

KEYBOARD PORT ASSIGNMENTS

Keyboard Status Port=STAPT=FA
Status Bit=KDR=bit Ø (LOW means READY).
Keyboard Data Port=KDATA=FC (8 bits, normal, unstripped).

NOTE: Status Port FA is also used by the Cassette Interface, and the Parallel Port.

The following is the SOL KEYBOARD INPUT ROUTINE

```
KSTAT   IN STAPT        ;GET STATUS WORD.
        CMA             ;INVERT IT FOR PROPER RETURN.
        ANI KDR         ;TEST ONLY KEYBOARD BIT.
        RZ              ;RETURN WITH "Z" SET IF EMPTY.
        IN KDATA        ;GET CHARACTER.
        RET             ;*** UNSTRIPPED CHARACTER IN "A" ***
```

1: SERIAL PORT: SSTAT: CØ42

Character returned in "A": "Z" flag set if no character found: Only the "A" register affected. No stripping is done of the MSB (Most Significant Bit). If a character was found, status is reset.

The Serial Port is pseudoport 1. Besides being available via calls to SINP and AINP, it may be directly accessed via a CALL SSTAT (CALL CØ42). It is suggested that either SINP or AINP (with "A" set to 1), be used to access this routine, as these are standard entry points for all SOL systems, but the address of SSTAT may vary.

The SOL interface connector for the SERIAL port is connected as if the SOL were itself a terminal, not a computer. This can cause serious problems when connecting to most other terminals, since most people use their SOL as a computer rather than as a terminal. The following pairs of pins on the serial connector should be EXCHANGED:

2 and 3	Transmit and Receive
4 and 5	Request To Send and Clear To Send
6 and 2Ø	Data Set Ready and Data Terminal Ready

The above are EIA RS232C signals. If using a current loop interface, no exchanges are necessary.

It is also interesting to note that many printers can communicate adequately with only 1,2,3, and 7 connected, since not all serial terminals supply or need handshaking signals.

I perform the necessary exchanges by having a short cable with a male on one end, a female on the other, and only the lines 1,2,3,4,5,6,7, and 2Ø connected (with the appropriate switchovers being made at the male connector). This method requires no changes to either the SOL or the terminal, and also provides a longer connector length overall.

J1 PINOUTS FOR SERIAL CONNECTOR (FEMALE DB25S ON SOL)

1	CG	CHASSIS GROUND
2	TD	TRANSMIT DATA (OUT)
3	RD	RECEIVE DATA (IN)
4	RTS	REQUEST TO SEND (OUT)
5	CTS	CLEAR TO SEND (IN)
6	DSR	DATA SET READY (IN)
7	SG	SIGNAL GROUND
8	CD	CARRIER DETECT (IN)
11	CLO	CURRENT LOOP OUT
12	LR1	LOOP RECEIVER 1
13	LR2	LOOP RECEIVER 2
2Ø	DTR	DATA TERMINAL READY (OUT)
23	LCS	LOOP CURRENT SOURCE

The following port information is made available for those who might find it useful.
Serial Status Port=SERST=F8
Status Bit=SDR=bit 6 (HIGH means READY)
Serial Data Port=SDATA=F9 (8 bits, normal, unstripped).
*** beware of parity bits! Strip off MSB ***

Additional status bits associated with the SERIAL port, but not handled in the SOL software are shown below.

BIT	NAME	FUNCTION	ACTIVE DIRECTION
Ø	SCD	Serial Carrier Detect (EIA)	1=Carrier on
1	SDSR	Serial Data Set Ready (EIA)	Ø=Link OK
2	SPE	Serial Parity Error (UART)	1=Error
3	SFE	Serial Framing Error (UART)	1=Error
4	SOE	Serial Overrun Error (UART)	1=Error
5	SCTS	Serial Clear To Send (EIA)	Ø=Clear
6	SDR	Serial Data Ready (UART)	1=Ready
7	STBE	Serial Xmit Buff Empty (UART)	1=Empty

The SOL SERIAL INPUT ROUTINE follows:

```
SSTAT   IN SERST        ;GET SERIAL STATUS WORD.
        ANI SDR         ;TEST FOR SERIAL DATA READY.
        RZ              ;RETURN WITH "Z" SET IF NO CHARACTER.
        IN SDATA        ;GET INPUT CHARACTER.
        RET             ;*** CHARACTER IN "A", "Z" IS RESET ***
```

2: PARALLEL PORT: PASTAT: C2DD

Character returned in "A": "Z" flag set if no character found: Only the "A" register affected. No stripping is done of the MSB (Most Significant Bit). If a character was found, status is reset.

The Parallel Port is pseudoport 2. Besides being available via calls to SINP and AINP, it may be directly accessed via a CALL PASTAT (CALL C2DD). It is suggested that either SINP or AINP (with "A" set to 1), be used to access this routine, as these are standard entry points for all SOL systems, but the address of PASTAT may vary.

PARALLEL PORT ASSIGNMENTS

Parallel Port Status=STAPT=FA
Status Bit=PDR=bit 1 (LOW means READY)
Parallel Data Port=PDATA=FD

NOTE: Status Port FA is also used by the Keyboard and Cassete Interface.

J2 PINOUTS FOR PARALLEL CONNECTOR (MALE DB25P ON SOL)

1	CG	CHASSIS GROUND		
2	SG	SIGNAL GROUND		
3	IE	INPUT ENABLE	(IN)	1=ENABLED
4	*DR	*DATA READY	(IN)	Ø=READY
5	*IAK	*INPUT ACKNOWLEDGE	(OUT)	Ø=ACKNOWLEDGE
6	ID7	INPUT DATA, BIT 7	(IN)	
7	ID6	INPUT DATA, BIT 6	(IN)	
8	ID5	INPUT DATA, BIT 5	(IN)	
9	ID4	INPUT DATA, BIT 4	(IN)	
1Ø	ID3	INPUT DATA, BIT 3	(IN)	
11	ID2	INPUT DATA, BIT 2	(IN)	
12	ID1	INPUT DATA, BIT 1	(IN)	
13	IDØ	INPUT DATA, BIT Ø	(IN)	

CONTINUED ON PAGE 4.

UNDERSTANDING...CONTINUED FROM PAGE 3.

14	US	UNIT SELECT	(OUT)	
15	OE	OUTPUT ENABLE	(IN)	1=ENABLED
16	*XDR	*EXTERNAL DEVICE READY	(IN)	0=READY
17	*OL	*OUTPUT LOAD	(OUT)	0=LOAD
18	OD7	OUTPUT DATA, BIT 7	(OUT)	
19	OD6	OUTPUT DATA, BIT 6	(OUT)	
20	OD5	OUTPUT DATA, BIT 5	(OUT)	
21	OD4	OUTPUT DATA, BIT 4	(OUT)	
22	OD3	OUTPUT DATA, BIT 3	(OUT)	
23	OD2	OUTPUT DATA, BIT 2	(OUT)	
24	OD1	OUTPUT DATA, BIT 1	(OUT)	
25	OD0	OUTPUT DATA, BIT 0	(OUT)	

Many of the INPUT lines have active pullup. For the simplest case Parallel Input with HANDSHAKING, the following hardware arrangement can be used.

1,2 BOTH connected to Parallel Input Device GROUND.
3 Leave floating HIGH.
4 Parallel Input Device must supply ACTIVE LOW signal.
(Should go LOW when DATA is ready to be sent).
5 Parallel Input Device must sample this line.
(Whenever it is LOW, Data MAY be sent).
6-13 Parallel Input Device should supply DATA on these lines.

The following is the SOL PARALLEL INPUT ROUTINE

```
PASTAT  IN STAPT        ;GET STATUS WORD.
        CMA             ;INVERT FOR PROPER RETURN.
        ANI PDR         ;TEST PARALLEL PORT ONLY.
        RZ              ;RETURN WITH "Z" SET IF EMPTY.
        IN PDATA        ;GET PARALLEL DATA.
        RET             ;*** UNSTRIPPED CHARACTER IN "A" ***
```

3 USER-DEFINED PORT: ERRIT: C2CB

Character returned in "A": "Z" flag set if no character found: Only the "A" register affected. If a character was found, status is reset. Any OTHER characteristics are up to the USER, but THESE CHARACTERISTICS MUST BE MET!

The CUSTOM or USER-DEFINED PORT is pseudoport 3. Besides being available via calls to SINP and AINP, it may be directly accessed via a CALL ERRIT (CALL C2CB), or by means of a direct call to the address of the user routine. It is suggested that either SINP or AINP (with "A" set to 3), be used to access this routine, as these are standard entry points for all SOL systems, but the address of ERRIT may vary.

NOTE: Use of pseudoport 3 assumes that the user has placed an input routine in memory that meets the above requirements. In addition, the address of the CUSTOM routine must be loaded into memory at **UIPRT (C800-C801)**. There are two ways of accomplishing this: The user may type SE CI=XXXX, where XXXX is the address of the user-defined CUSTOM routine, or he may load the address in under program control. For example, the instruction sequence LXI H,1234 / SHLD C800 would cause the address 1234 to be loaded at C800-C801. (Addresses are loaded in reverse-order... Low order at C800, and High order at C801).

Remember also that it is up to the user to make the current pseudoport=3 when using SINP, and to make "A"=3 when using AINP to access the CUSTOM Input Routine. After a RESET, the IPORT and UIPRT are set to 00 and 0000 repectively. The ERRIT routine will assume an error has occurred whenever it finds 0000 at UIPRT. This means that 0000 may never be used as the address of a CUSTOM routine.

The CUSTOM input routine allows the user to write input routines to handle special devices, and even include special character checks and conversions within the CUSTOM routine. The user can often put this ability to good use. I have, for instance, written CUSTOM input drivers that will take programs written for one version of BASIC and convert them to another version. Also note that it is often useful to have certain CUSTOM routines automatically switch the current pseudoport to 0 when a given process has been completed. The CUSTOM routines are often much more than just simple input drivers! Their use is limited only by the imagination and skill of the user.

More than one custom routine may reside in memory at once, but only one custom routine at a time may be called in the flexible manner allowed by SINP and AINP. This is not really a problem, since control characters may be used to allow one CUSTOM routine to make another custom routine current. I routinely use CONTROL/X as a special character that causes any given CUSTOM routine to load in another custom routine. You have to design your own CUSTOM routines to meet your own particular needs, but believe me, the ability to access such programs using SINP and AINP is a powerful feature.

This documentation provided courtesy of
Fr. Thomas McGahee
Don Bosco Tech
202 Union Ave.
Paterson, NJ 07502 phone (201) 595-8800

LOADING AND SAVING MICROPOLIS SOURCE/TEXT FILES ON SOLOS CASSETTES --- VERSION 2.1

By Melvin M. Dalton

INTRODUCTION

This program is provided for SOL-20 users who also have one or more MICROPOLIS Mod I/II drives. The program assume you are using MICROPOLIS PDS VS 4.0 (1978). The assembly language program resides in the MDOS Application Program area starting at 2B00H. It is designed to be called as an implicit command from MDOS.

The program will read or write directly between a cassette and the disk. The format of the cassette tape may be either PROTEUS Library standard byte mode or ALS8 format. The choice of format and whether to read or write on the disk is a user option.

This program completes the set for MICROPOLIS users. The programs published in PROTEUS NEWS Vol. 2 No. 4 provide for byte mode cassette reading and writing of BASIC programs and this one does the same for assembly source or plain text files.

ASSUMPTIONS

1. PROTEUS Library byte mode format is:
   <4 byte line number> <20H> <text> <CR>
2. ALS8/SCS16 block mode format is:
   <line length> <4 byte line number> <20H> <text> <CR>
3. All text is ASCII (20H through 7EH).
4. All control characters are ignored except <CR>.
5. This program generates cassette files with exactly the format listed above.

CONTINUED ON NEXT PAGE.

6. This program will read cassette files with the following format errors:
   a. Byte mode files may have from 1 to 4 bytes of line number. Line numbers of 5 or more bytes are loaded but produce an error message to warn the operator of possible editing needs.
   b. Byte mode files may have no space (20H) between the line number and the text.
   c. ALS8/SCS16 block mode files MUST be exactly as shown above except the space after the line number may be omited IF there is no text.
7. Loading from a tape with more than 132 characters between <CR>'s will be aborted with an appropriate error message.

PREPARING THE PROGRAM FOR USE

Either of the following methods may be used:

A. Simple and fast:
   1. From PROTEUS library cassette GET MSCSA (object code file)
   2. Boot MDOS and SAVE "CASSETTE" 2B00 2E61 18
   3. CASSETTE is now an implicit MDOS command.

B. Slow but sure:
   1. Boot MDOS and enter LINEEDIT
   2. Enter source text given below (watch those typo's!!).
   3. Be sure you have source file called SOLEQU, if not, copy from below and SAVE "SOLEQU".
   4. Assemble program, correct any typo's, and execute a TYPE "CASSETTE" 18 command.
   5. CASSETTE is now an implicit MDOS command.

SYNTAX OF IMPLICIT COMMAND CASSETTE

```
CASSETTE "[drive #:]<diskfile name>"
               "<SOL file name>" "<option>"
```

[drive #:] must be 0-3 (default is 0)
"<SOL file name>" may not exceed five characters
option "RA" means read disk & write ALS8 format tape
option "RB" means read disk & write byte mode tape
option "WA" means read ALS8 format tape & write on disk
option "WB" means read byte mode tape & write on disk

EXAMPLES:

CASSETTE "1:CASSETTES" "CASS7" "WB" --- this command will create an assembly source file called "CASSETTES" on drive #1 from the byte-mode source/text cassette file called CASS7.

CASSETTE "TEST" "MMD" "RA" --- this command will make an ALS8 format cassette recording called MMD of the MDOS type 4-7 source/text file found on drive #0 and called "TEST"

The program will sign on with the message:

```
CASSETTE UTILITY --- MDOS/SOLOS  VERSION 2.1   11/79
```

ERROR MESSAGES

All MDOS error messages have their standard meanings and all error messages generated by CASSETTE are in plain English. However there is one which may warrant further explaination. During a "WB" option ONLY the message:

```
LINE # FORMAT CORRECTED. RENUMBER & EDIT BEFORE USING
```

will appear just before END-FILE if:

1. The line number is five or more bytes long. In this case the excess bytes will become part of the text (where they will need to be deleted) and the truncated line numbers may be out of numerical order. RENUM before editing.
2. The space <20H> is missing between the line number bytes and the text. This case is treated the same as 1. above by the program. RENUM and check if any editing is needed.

LISTING OF SOURCE FILE "SOLEQU"

```
0000                NLIST
0010 IPORT          EQU    0C806H
0020 OPORT          EQU    0C807H
0030 UIPORT         EQU    0C800H
0040 UOPORT         EQU    0C802H
0050 USARE          EQU    0CAB4H
0060 FCBA11         EQU    0C855H
0070 FCBA12         EQU    0C85CH
0080 FBUF11         EQU    0C863H
0090 FBUF12         EQU    0C963H
0100 SCREEN         EQU    0CC00H
0110 CRLF           EQU    0C2F9H
0120 START          EQU    0C000H
0130 INIT           EQU    0C001H
0140 RETRN          EQU    0C004H
0150 FOPEN          EQU    0C007H
0160 FCLOS          EQU    0C00AH
0170 RDBYT          EQU    0C00DH
0180 WRBYT          EQU    0C010H
0190 RDBLK          EQU    0C013H
0200 WRBLK          EQU    0C016H
0210 SOUT           EQU    0C019H
0220 AOUT           EQU    0C01CH
0230 SINP           EQU    0C01FH
0240 AINP           EQU    0C022H
0250 ADOUT          EQU    0C3E8H
0260 THEAD          EQU    0C81CH
0270 DHEAD          EQU    0C82CH
0280                LIST
```

```
0000            *
0000            **********************************
0000            *** PROGRAM TO TRANSFER CUTS BYTE
0000            *** MODE FILES TO/FROM MICROPOLIS
0000            *** SOURCE/TEXT FILES.
0000            **********************************
0000            *
0000            *** WRITTEN BY MELVIN M. DALTON
0000            *** VERSION 2.1   NOVEMBER 1979
0000            *
0000                     TAB  10,15,27
0000                     LINK 'SYSQ1'
0000                     LINK 'SYSQ2'
0000                     LINK 'SOLEQU'
0000                     ORG  @APROGRAM
2B00            *
2B00            *   PARSE COMMAND LINE WITH FOLLOWING FORMAT:
2B00            *   CASSETTE "[drive #:]<diskfile name>"
2B00            *                  "<SOL file name>" "<option>"
2B00            *   [drive #:] MUST BE 0-3 (DEFAULT IS 0)
2B00            *   "<SOL file name>" MAY NOT EXCEED FIVE CHAR.
2B00            *   OPTION "RB" = READ DISK & WRITE BYTE MODE TAPE
2B00            *   OPTION "WB" = WRITE ON DISK FROM BYTE MODE TAPE
2B00            *   OPTION "RA" = READ DISK & WRITE ALS8 TAPE
2B00            *   OPTION "WA" = WRITE ON DISK FROM ALS8 TAPE
2B00            *
```

MICROPOLIS...CONTINUED. Pg 6

MICROPOLIS...CONTINUED.

```
2B00 CD 0D 08   CASSETTE   CALL @CCRLF
2B03 3E 05                 MVI  A,5          ;MESSAGE FIVE
2B05 21 A9 2D              LXI  H,MESSAGES
2B08 CD C6 1C              CALL @MESSAGEOUT ;SIGN-ON TO CRT
2B0B CD 0D 08              CALL @CCRLF
2B0E 21 A0 01              LXI  H,@INBUFF    ;POINT TO COMMAND LINE
2B11 0E 20                 MVI  C,' '        ;SPACE
2B13 CD 22 1C              CALL @SCAN        ;SEARCH FOR SPACE
2B16 DA 74 2C              JC   SYNERR
2B19 CD 1A 1C              CALL @SKIPSPACE
2B1C DA 74 2C              JC   SYNERR
2B1F CD 49 1F              CALL @PARAM       ;SEPARATE PARAMETERS
2B22 DA 8F 1C              JC   @DISKERROR
2B25 3A AC 29              LDA  @NASCPAR     ;GET NO. OF ASCII
2B28                                         ;PARAMETERS
2B28 FE 03                 CPI  3            ;SHOULD BE THREE
2B2A C2 74 2C              JNZ  SYNERR       ;OUTPUT IF IN ERROR
2B2D            *
2B2D            *   CHECK OPTION AND BRANCH
2B2D            *
2B2D 21 C3 29              LXI  H,@ASCBUFF2 ;OPTION BUFFER
2B30 7E                    MOV  A,M          ;GET 1ST BYTE
2B31 23                    INX  H            ;NEXT BYTE
2B32 86                    ADD  M            ;ADD IT
2B33 FE 93                 CPI  'R'+'A'      ;READ ALS8
2B35 CA 62 2B              JZ   READALS
2B38 FE 94                 CPI  'R'+'B'      ;READ BYTE MODE
2B3A CA 4A 2B              JZ   READBYTE
2B3D FE 98                 CPI  'W'+'A'      ;WRITE ALS8
2B3F CA 9B 2B              JZ   WRITEALS
2B42 FE 99                 CPI  'W'+'B'      ;WRITE BYTE MODE
2B44 CA 56 2B              JZ   WRITEBYTE
2B47 C3 5E 2D              JMP  OPTERR       ;OPTION ERROR
2B4A            *
2B4A            *   READ FROM DISK TO CUTS BYTE MODE
2B4A            *
2B4A CD F4 2B   READBYTE   CALL READDISK     ;DISK RDY TO READ
2B4D 21 1C C8              LXI  H,THEAD      ;POINT TO TAPE HEADER
2B50 CD 19 2C              CALL BYTEOPEN     ;TAPE RDY TO USE
2B53 C3 30 2C              JMP  READLOOP     ;GO DO IT
2B56            *
2B56            *   WRITE ON DISK FROM CUTS BYTE MODE
2B56            *
2B56 CD 80 2C   WRITEBYTE CALL WRITEDISK    ;DISK RDY TO RCD
2B59 21 2C C8              LXI  H,DHEAD      ;POINT TO DUMMY HEADER
2B5C CD 19 2C              CALL BYTEOPEN     ;TAPE RDY TO USE
2B5F C3 97 2C              JMP  WRITELOOP    ;GO DO IT
2B62            *
2B62            *   READ FROM DISK TO ALS8 BLOCK MODE TAPE
2B62            *
2B62 CD F4 2B   READALS    CALL READDISK     ;GET DISK READY
2B65 21 1C C8              LXI  H,THEAD      ;FILE NAME
2B68 CD 5D 2C              CALL TRANSFER     ;TO HEADER
2B6B 3E C1                 MVI  A,'A'+80H    ;TYPE
2B6D 32 22 C8              STA  THEAD+6
2B70 21 00 30              LXI  H,3000H      ;MAIN BUFFER
2B73 11 00 00              LXI  D,0          ;ZERO COUNTER
2B76 22 25 C8              SHLD THEAD+9      ;LOAD ADDR TO HDR
2B79 06 01      ALSRD      MVI  B,1          ;DISK FILE #1
2B7B CD 7A 18              CALL @RFINXPOSI   ;GET BYTE
2B7E DA 87 2B              JC   DONE
2B81 71                    MOV  M,C          ;IN MAIN BUFFER
2B82 23                    INX  H
2B83 13                    INX  D
2B84 C3 79 2B              JMP  ALSRD        ;RPT. TILL DONE
2B87 FE 02      DONE       CPI  2            ;EOF ?
2B89 C2 8F 1C              JNZ  @DISKERROR   ;ERROR
2B8C EB                    XCHG              ;GET COUNT IN HL
2B8D 22 23 C8              SHLD THEAD+7      ;CNT IN HDR
2B90 21 1C C8              LXI  H,THEAD
```

```
2B93 3E 80                 MVI  A,80H        ;1200 BAUD, TAPE 1
2B95 CD 16 C0              CALL WRBLK        ;PUT ON TAPE
2B98 C3 59 2D              JMP  DISKEOF
2B9B            *
2B9B            *   WRITE ON DISK FROM ALS8 BLOCK MODE TAPE
2B9B            *
2B9B CD 80 2C   WRITEALS   CALL WRITEDISK    ;GET DISK READY
2B9E 21 2C C8              LXI  H,DHEAD      ;DUMMY HEADER
2BA1 E5                    PUSH H            ;SAVE IT
2BA2 CD 5D 2C              CALL TRANSFER     ;MOVE NAME
2BA5 E1                    POP  H            ;PNT TO TAPE HDR
2BA6 11 00 32              LXI  D,3200H      ;TAPE LOAD ADDR
2BA9 3E 80                 MVI  A,80H        ;1200 BAUD, TAPE 1
2BAB CD 13 C0              CALL RDBLK        ;READ TAPE
2BAE DA 6B 2D              JC   READERR      ;READ ERROR
2BB1            *
2BB1            *   MOVE DATA WHILE CHECKING
2BB1            *   FOR LINES WITH FOUR
2BB1            *   BYTE LINE NUMBERS FOLLOWED
2BB1            *   BY <CR> WITHOUT SPACE (20H)
2BB1            *
2BB1 21 00 30              LXI  H,3000H      ;DESTINATION
2BB4 11 00 32              LXI  D,3200H      ;SOURCE
2BB7 1A         MOVE       LDAX D            ;GET LENGTH BYTE
2BB8 FE 85                 CPI  133          ;LINE TO LONG
2BBA D2 75 2D              JNC  LINEERR      ;FOR MDOS
2BBD FE 01                 CPI  1            ;EOF
2BBF CA CE 2B              JZ   FINISH
2BC2 FE 06                 CPI  6            ;MISSING SPACE
2BC4 CA E2 2B              JZ   FIXSPC
2BC7 4F                    MOV  C,A          ;LINELENGTH
2BC8 CD 4C 1B              CALL @TRANSDHC    ;MOVE LINE
2BCB C3 B7 2B              JMP  MOVE         ;REPEAT TILL FINISH
2BCE EB         FINISH     XCHG              ;DEST.+LENGTH IN DE
2BCF 21 00 30              LXI  H,3000H      ;START OF BLOCK
2BD2 CD 9E 1B              CALL @DESUBHL     ;[DE]-[HL]=[BC]
2BD5 50                    MOV  D,B          ;PUT BLOCK
2BD6 59                    MOV  E,C          ;LENGTH IN DE
2BD7 06 01                 MVI  B,1          ;DISK FILE 1
2BD9 CD 5E 19              CALL @SAVEDATA    ;PUT ON DISK
2BDC DA 8F 1C              JC   @DISKERROR
2BDF C3 4E 2D              JMP  EOFERROR1    ;PUT EOF ON
2BE2                                         ;DISK & QUIT
2BE2 3E 07      FIXSPC     MVI  A,7          ;NEW LINE LENGTH
2BE4 12                    STAX D            ;IN SOURCE
2BE5 0E 05                 MVI  C,5          ;MOVE ALL
2BE7 CD 4C 1B              CALL @TRANSDHC    ;BUT <CR>
2BEA 13                    INX  D            ;POINT TO NEW
2BEB                                         ;LINE IN SOURCE
2BEB 36 20                 MVI  M,20H        ;SPACE
2BED 23                    INX  H            ;IN DESTINATION
2BEE 36 0D                 MVI  M,0DH        ;<CR>
2BF0 23                    INX  H            ;IN DESTINATION
2BF1 C3 B7 2B              JMP  MOVE         ;NEXT LINE
2BF4            *
2BF4            *   PREPARE TO READ FROM DISK
2BF4            *
2BF4 0E 00      READDISK   MVI  C,0          ;POINT TO ASCBUFF0
2BF6 CD 6B 1B              CALL @TRANSFILENAME ;MOVE ASCBUFF0
2BF9                                         ;TO ASCIIBUFF
2BF9 06 01                 MVI  B,1          ;LOGICAL FILE #1
2BFB 3A D7 29              LDA  @DRIVENO     ;DRIVE # FOR DISK
2BFE 4F                    MOV  C,A          ;PUT IN "C"
2BFF 21 4B 26              LXI  H,@FILEBUFFER0 ;USE DISK BUFFER #0
2C02 CD 09 17              CALL @OPENFILE
2C05 DA 8F 1C              JC   @DISKERROR
2C08 CD 91 17              CALL @RFILEINF    ;GET FILE INFO
2C0B DA 8F 1C              JC   @DISKERROR
2C0E 78                    MOV  A,B          ;PUT IN "A" FOR TEST
2C0F E6 FC                 ANI  0FCH         ;MAKE IT MODULO 4
```

CONTINUED Pg.7

CONTINUED

```
2C11 FE 04                CPI   4             ;IS IT TYPE 4?
2C13 3E 11                MVI   A,17          ;WRONG FILE TYPE
2C15 C2 8F 1C             JNZ   @DISKERROR    ;MESSAGE OUT
2C18 C9                   RET
2C19            *
2C19            *   PREPARE TAPE FOR BYTE MODE
2C19            *
2C19 E5         BYTEOPEN  PUSH H              ;SAVE HDR PNTR ON STACK
2C1A CD 5D 2C             CALL TRANSFER       ;MOVE THE NAME
2C1D 3E C1                MVI   A,'A'+80H
2C1F 32 22 C8             STA   THEAD+6       ;SOL TAPE TYPE
2C22 21 2B 2C             LXI   H,$+9         ;CORRECT FOR STACK
2C25                                          ;ERROR IN SOLOS 1.3
2C25 E3                   XTHL                ;POINT TO HEADER
2C26 3E 01                MVI   A,1           ;TAPE FILE #1
2C28 CD 07 C0             CALL FOPEN          ;OPEN FILE
2C2B DA 66 2D             JC    OPENERR
2C2E E1                   POP   H             ;IF NO ERROR RETURN
2C2F                                          ;STACK TO ORIGINAL
2C2F                                          ;STATE
2C2F C9                   RET
2C30            *
2C30            *   MAIN TRANFER LOOP
2C30            *   DISK TO TAPE
2C30            *   DELETE LINE LENGTH
2C30            *
2C30 06 01      READLOOP  MVI   B,1           ;POINT TO FILE #1
2C32 CD 7A 18             CALL @RFINXPOSI     ;THROW FIRST BYTE OUT
2C35 DA 93 2D             JC    EXIT2
2C38 0D                   DCR   C             ;IS LENGTH = 01?
2C39 CA 59 2D             JZ    DISKEOF       ;END OF FILE
2C3C 21 A0 01             LXI   H,@INBUFF
2C3F 59                   MOV   E,C           ;LENGTH IN
2C40 16 00                MVI   D,0           ;'DE'
2C42 CD CA 18             CALL @LOADDATA      ;GET LINE
2C45 DA 93 2D             JC    EXIT2
2C48 46         WRITETAPE MOV   B,M           ;GET BYTE
2C49 3E 01                MVI   A,1           ;TAPE FILE #1
2C4B E5                   PUSH H
2C4C D5                   PUSH D
2C4D CD 10 C0             CALL WRBYT          ;PUT BYTE ON TAPE
2C50 D1                   POP   D
2C51 E1                   POP   H
2C52 DA 70 2D             JC    WRITERR
2C55 23                   INX   H             ;POINT NEXT BYTE
2C56 1D                   DCR   E             ;COUNT LENGTH
2C57 C2 48 2C             JNZ   WRITETAPE     ;RECORD LINE
2C5A C3 30 2C             JMP   READLOOP      ;NEXT LINE
2C5D            *
2C5D            *   ROUTINE TO MOVE SOL
2C5D            *   FILE NAME FROM COMMAND
2C5D            *   LINE TO DESIRED HEADER.
2C5D            *   FILL WITH 00H TO MAKE
2C5D            *   FIVE CHARACTERS OR OUTPUT
2C5D            *   ERROR MESSAGE IF NAME
2C5D            *   TO LONG
2C5D            *
2C5D 11 B9 29   TRANSFER  LXI   D,@ASCBUFF1 ;TAPE NAME
2C60 0E 05                MVI   C,5           ;SOL NAMES <=5
2C62 1A         TRANSFER1 LDAX D              ;GET CHARACTER
2C63 FE 20                CPI   20H           ;IS IT 'SPACE'
2C65 C2 69 2C             JNZ   TRANSFER2     ;IF SO, CHANGE TO 0
2C68 AF                   XRA   A
2C69 77         TRANSFER2 MOV   M,A           ;PUT IN HEADER
2C6A 23                   INX   H
2C6B 13                   INX   D
2C6C 0D                   DCR   C             ;MAX NO. CHAR.
2C6D C2 62 2C             JNZ   TRANSFER1     ;NEXT CHARACTER
2C70 1A                   LDAX D              ;GET CHARACTER
2C71 FE 20                CPI   20H           ;IS IT 'SPACE'
2C73 C8                   RZ                  ;DONE
2C74 AF         SYNERR    XRA   A             ;SYNTAX ERROR
2C75 C3 8F 1C             JMP   @DISKERROR
```

```
2C78            *
2C78            *   TEST IF CHARACTER IS NUMBER
2C78            *   NUMBER = C FLAG SET
2C78            *   OTHERS = NC FLAG SET
2C78            *
2C78 FE 40      NUMBCHK   CPI   40H
2C7A F0                   RP                  ;NOT A NUMBER
2C7B FE 30                CPI   30H
2C7D 3F                   CMC                 ;REVERSE CARRY
2C7E D8                   RC                  ;A NUMBER
2C7F C9                   RET
2C80            *
2C80            *   PREPARE TO WRITE ON DISK
2C80            *
2C80 0E 00      WRITEDISK MVI   C,0           ;POINT TO ASCBUFF0
2C82 CD 6B 1B             CALL @TRANSFILENAME ;ASCBUFF0 TO
2C85                                          ;ASCIIBUFF
2C85 06 01                MVI   B,1           ;LOGICAL FILE #1
2C87 16 04                MVI   D,4           ;TYPE 4 FILE
2C89 3A D7 29             LDA   @DRIVENO      ;GET DRIVE #
2C8C 4F                   MOV   C,A           ;PUT IN 'C'
2C8D 21 4B 26             LXI   H,@FILEBUFFER0 ;USE DISK BUFFER #0
2C90 CD 6C 16             CALL @CREATE        ;NEW DISK FILE
2C93 DA 8F 1C             JC    @DISKERROR
2C96 C9                   RET
2C97            *
2C97            *   MAIN TRANSFER LOOP
2C97            *   FROM TAPE TO DISK
2C97            *   INSERT LINE LENGTHS
2C97            *
2C97 3E 30      WRITELOOP MVI   A,30H         ;ZERO IN 'A'
2C99 06 04                MVI   B,4           ;FOUR BYTES
2C9B 11 06 00             LXI   D,6           ;MINIMUM LINE LENGTH
2C9E 21 A1 01             LXI   H,@INBUFF+1 ;START OF TEXT
2CA1 CD 89 1B             CALL @FILLA         ;DEFAULT LINE # = 0000
2CA4 2B                   DCX   H             ;POINT LSB LINE #
2CA5 CD F0 2C   SKIP      CALL TAPE           ;GET BYTE
2CA8 CD 78 2C             CALL NUMBCHK        ;SKIP
2CAB D4 DA 2C             CNC   SETFLAG       ;SET NZ TO
2CAE                                          ;FLAG MESS.
2CAE D2 A5 2C             JNC   SKIP          ;IF NOT #
2CB1 77                   MOV   M,A           ;1ST BYTE ALWAYS #
2CB2 0E 03                MVI   C,3           ;REPEAT 3 TIMES
2CB4 CD F0 2C   WRTLOOP1  CALL TAPE
2CB7 CD 78 2C             CALL NUMBCHK
2CBA D2 C8 2C             JNC   NUMBDONE      ;NOT NUMBER
2CBD CD 18 2D             CALL SHFTLFT
2CC0 77                   MOV   M,A
2CC1 0D                   DCR   C             ;COUNT
2CC2 C2 B4 2C             JNZ   WRTLOOP1      ;AND REPEAT
2CC5 CD F0 2C             CALL TAPE
2CC8 23         NUMBDONE  INX   H             ;POINT TO LOCATION
2CC9                                          ;FOR SPACE
2CC9 FE 20                CPI   20H           ;IS IT SPACE?
2CCB C4 D6 2C             CNZ   INSRTSPC
2CCE FE 0D                CPI   0DH           ;IS IT <CR> ?
2CD0 CA 08 2D             JZ    BUFFDISK      ;PUT ON DISK
2CD3 C3 E4 2C             JMP   TAPEBUFF1     ;GET REST OF LINE
2CD6 36 20      INSRTSPC  MVI   M,20H         ;PUT SPACE IN
2CD8 23                   INX   H             ;POINT NXT LOC
2CD9 1C                   INR   E             ;INCREMENT LINELENGTH
2CDA 32 61 2E   SETFLAG   STA   LNFLAG        ;SET LINE NUMBER
2CDD                                          ;FLAG NON-ZERO
2CDD C9                   RET
2CDE            *
2CDE            *   MOVE REST OF ONE LINE OF
2CDE            *   TEXT FROM TAPE TO BUFFER
2CDE            *
2CDE CD F0 2C   TAPEBUFF  CALL TAPE           ;GET BYTE
2CE1 DA 08 2D             JC    BUFFDISK      ;PUT LINE ON
2CE4                                          ;DISK IF <CR>
2CE4 77         TAPEBUFF1 MOV   M,A           ;BYTE IN BUFFER
```

CONTINUED PAGE 8.

MICROPOLIS...CONTINUED.

```
2CE5 23                  INX  H
2CE6 1C                  INR  E             ;LENGTH COUNT
2CE7 7B                  MOV  A,E           ;IN "A" FOR TEST
2CE8 FE 85               CPI  133           ;MAXIMUM LINE LENGTH
2CEA D2 75 2D            JNC  LINEERR       ;LINE TO LONG
2CED C3 DE 2C            JMP  TAPEBUFF
2CF0           *
2CF0           *   GET ONE BYTE OF TEXT
2CF0           *   SKIP ALL CTRL'S
2CF0           *   EXCEPT <CR>
2CF0           *   NORMAL RETURN = NC
2CF0           *   <CR> RETURN = C
2CF0           *
2CF0 3E 01     TAPE      MVI  A,1           ;TAPE FILE #1
2CF2 E5                  PUSH H
2CF3 D5                  PUSH D
2CF4 C5                  PUSH B
2CF5 CD 0D C0            CALL RDBYT         ;GET TAPE CHARACTER
2CF8 C1                  POP  B
2CF9 D1                  POP  D
2CFA E1                  POP  H
2CFB DA 3C 2D            JC   EOFERROR      ;EOF OR ERROR
2CFE FE 20               CPI  20H           ;IS IT NOT CTRL?
2D00 F0                  RP                 ;RETURN IF NOT
2D01 FE 0D               CPI  0DH           ;IS IT <CR>?
2D03 C2 F0 2C            JNZ  TAPE          ;GET NEW BYTE
2D06 37                  STC
2D07 C9                  RET                ;<CR>=CARRY SET
2D08           *
2D08           *   RECORD LINE ON DISK
2D08           *
2D08 77        BUFFDISK  MOV  M,A           ;INCLUDE <CR>
2D09 21 A0 01            LXI  H,@INBUFF
2D0C 73                  MOV  M,E           ;PUT IN FIRST SLOT
2D0D 06 01               MVI  B,1           ;DISK FILE #1
2D0F CD 5E 19            CALL @SAVEDATA     ;PUT LINE ON DISK
2D12 DA 9B 2D            JC   EXIT3
2D15 C3 97 2C            JMP  WRITELOOP     ;DO IT AGAIN
2D18           *
2D18           *   SHIFT LINE NUMBER LEFT
2D18           *   ONE BYTE WHEN CALLED
2D18           *
2D18 E5        SHFTLFT   PUSH H             ;SAVE "HL"
2D19 2A A1 01            LHLD @INBUFF+1     ;GET MSB &MSB+1
2D1C 22 A0 01            SHLD @INBUFF       ;ONE BYTE LEFT
2D1F 2A A3 01            LHLD @INBUFF+3     ;GET NEXT LSB & LSB
2D22 22 A2 01            SHLD @INBUFF+2     ;ONE BYTE LEFT
2D25 E1                  POP  H
2D26 C9                  RET
2D27           *
2D27           *   ERROR AND EXIT ROUTINES
2D27           *
2D27 3A D7 29  CLOSE     LDA  @DRIVENO      ;DRIVE #
2D2A 4F                  MOV  C,A           ;TO C
2D2B 06 01               MVI  B,1           ;FILE # TO B
2D2D CD 54 17            CALL @CLOSEFILE    ;DISK FILE
2D30 DC B1 1C            CC   @ERRORMES
2D33 3E 01               MVI  A,1           ;TAPE FILE #
2D35 CD 0A C0            CALL FCLOS
2D38 21 A9 2D            LXI  H,MESSAGES    ;USER MESS.
2D3B C9                  RET
2D3C           *
2D3C F2 6B 2D  EOFERROR  JP   READERR       ;ERROR
2D3F 3A 61 2E            LDA  LNFLAG        ;CHECK FOR LINE
2D42 A7                  ANA  A             ;NUMBER FORMAT ERRORS
2D43 CA 4E 2D            JZ   EOFERROR1     ;NONE
2D46 21 A9 2D            LXI  H,MESSAGES
2D49 3E 06               MVI  A,6           ;MESSAGE SIX
2D4B CD C6 1C            CALL @MESSAGEOUT
2D4E 21 5F 2E  EOFERROR1 LXI  H,EOFBUFF     ;POINT TO EOF DATA
2D51 11 02 00            LXI  D,2           ;SAVE TWO BYTES
2D54 06 01               MVI  B,1           ;DISK FILE #1
2D56 CD 5E 19            CALL @SAVEDATA     ;PUT ON DISK
```

```
2D59 3E 02     DISKEOF   MVI  A,2
2D5B C3 93 2D            JMP  EXIT2
2D5E 3E 00     OPTERR    MVI  A,0
2D60 21 A9 2D            LXI  H,MESSAGES
2D63 C3 C6 1C            JMP  @MESSAGEOUT
2D66 3E 01     OPENERR   MVI  A,1           ;POINT TO MESSAGE #1
2D68 C3 88 2D            JMP  EXIT1
2D6B 3E 02     READERR   MVI  A,2           ;POINT TO MESSAGE #2
2D6D C3 77 2D            JMP  EXIT0
2D70 3E 03     WRITERR   MVI  A,3           ;POINT TO MESSAGE #3
2D72 C3 88 2D            JMP  EXIT1
2D75 3E 04     LINEERR   MVI  A,4           ;POINT TO MESSAGE #4
2D77 F5        EXIT0     PUSH PSW           ;SAVE MESSAGE #
2D78 CD 27 2D            CALL CLOSE         ;SHUT FILES
2D7B CD 3C 16            CALL @SCRATCH      ;DISK FILE
2D7E DC B1 1C            CC   @ERRORMES     ;ERROR
2D81 F1                  POP  PSW           ;MESSAGE #
2D82 CD C6 1C            CALL @MESSAGEOUT
2D85 C3 0C 20            JMP  @MDOSEXECUTIVE
2D88 F5        EXIT1     PUSH PSW           ;SAVE MESS. #
2D89 CD 27 2D            CALL CLOSE
2D8C F1                  POP  PSW           ;GET MESS. #
2D8D CD C6 1C            CALL @MESSAGEOUT
2D90 C3 0C 20            JMP  @MDOSEXECUTIVE
2D93 F5        EXIT2     PUSH PSW           ;SAVE DISK ERROR
2D94 CD 27 2D            CALL CLOSE         ;SHUT FILES
2D97 F1                  POP  PSW           ;GET DISK ERROR #
2D98 C3 8F 1C            JMP  @DISKERROR
2D9B F5        EXIT3     PUSH PSW           ;SAVE MESS. #
2D9C CD 27 2D            CALL CLOSE         ;SHUT FILES
2D9F CD 3C 16            CALL @SCRATCH      ;DELETE FILE
2DA2 DC B1 1C            CC   @ERRORMES
2DA5 F1                  POP  PSW           ;GET MESS. #
2DA6 C3 8F 1C            JMP  @DISKERROR
2DA9           *
2DA9           *   ERROR MESSAGES
2DA9           *
2DA9 49 4C 4C  MESSAGES  DTH  'ILLEGAL OPTION'
2DAC 45 47 41
2DAF 4C 20 4F
2DB2 50 54 49
2DB5 4F CE
2DB7 54 41 50            DTH  'TAPE FILE OPEN'
2DBA 45 20 46
2DBD 49 4C 45
2DC0 20 4F 50
2DC3 45 CE
2DC5 54 41 50            DTH  'TAPE READ ERROR'
2DC8 45 20 52
2DCB 45 41 44
2DCE 20 45 52
2DD1 52 4F D2
2DD4 54 41 50            DTH  'TAPE WRITE ERROR'
2DD7 45 20 57
2DDA 52 49 54
2DDD 45 20 45
2DE0 52 52 4F
2DE3 D2
2DE4 54 45 58            DTH  'TEXT LINE TO LONG'
2DE7 54 20 4C
2DEA 49 4E 45
2DED 20 54 4F
2DF0 20 4C 4F
2DF3 4E C7
2DF5 43 41 53            DT   'CASSETTE UTILITY --- MDOS/SOLOS
2DF8 53 45 54
2DFB 54 45 20
2DFE 55 54 49
2E01 4C 49 54
2E04 59 20 2D
2E07 2D 2D 20
2E0A 4D 44 4F
2E0D 53 2F 53
```

CONTINUED Pg.9.

MICROPOLIS TEXT/SOURCE FILES   CONTINUED

```
2E10 4F 4C 4F
2E13 53 20 20
2E16 56 45 52               DTH  'VERSION 2.1   11/79'
2E19 53 49 4F
2E1C 4E 20 32
2E1F 2E 31 20
2E22 20 20 31
2E25 31 2F 37
2E28 B9
2E29 4C 49 4E               DT   'LINE # FORMAT CORRECTED.
2E2C 45 20 23
2E2F 20 46 4F
2E32 52 4D 41
2E35 54 20 43
2E38 4F 52 52
2E3B 45 43 54
2E3E 45 44 2E
2E41 20
2E42 52 45 4E               DTH  'RENUMBER & EDIT BEFORE USING.'
2E45 55 4D 42
2E48 45 52 20
2E4B 26 20 45
2E4E 44 49 54
2E51 20 42 45
2E54 46 4F 52
2E57 45 20 55
2E5A 53 49 4E
2E5D 47 AE
2E5F           *
2E5F           *   CONSTANTS
2E5F           *
2E5F 01 FF     EOFBUFF   DD   1FFH          ;EOF DATA
2E61 00        LNFLAG    DB   0             ;LINE NUMBER FLAG
2E62                                        ;INITIALIZED TO ZERO
2E62                     END  CASSETTE

ERRORS THIS ASSEMBLY  0000
```

Defining the Problem

QUESTIONS THAT NEED ANSWERS

8KRA'S AND POWER PROBLEMS IN NEPAL

I am a Medical Missionary in Nepal and use the Sol for word processing in the Nepali language. Of course it is fitted with a programmable character generator to be able to work in a language with a different script. I need answers to the follwing questions:

1. I have 8KRA memory boards and would like to put the jumper to have them protected on power up since we have so many power failures (blinks) here. I have the standby battery packs but find that random material is often written into the memory and causes the program to fail later. Since the Sol has no signals on S-100 20-70 for protect, unprotect, do you know of a software way to unprotect the boards when they are jumpered to come up protected?

2. I was advised not to bring disc systems to Nepal because they were too critical as to alignment etc. I was also told that since I could be operating in temperatures that would vary from 45F to 90F that I may find a program written in cold weather may not read in hot weather. This is fine! For long time storage I will continue to use tape since it will be more realiable, but since we have the power problems and have to now and again do restarts I wonder if a disc just for day to day would have that much trouble? Can you recommend this and if so what small single drive disc would you recommend for day to day development? It would not have to have too complicated an operating system since it would be mostly used just to restart the lost word processing system and for temporary storage. When the days work would be finished then all would be put on tape. Maybe you would recommend a higher speed tape system?

C. Dudley Henderson, D.D.S.
Box 1211
Kathmandu, Nepal

Editor: I live in an area that the ambient room temperature runs between 50F. and 110F. and have had no trouble with my North Star single drive system. My Nortn Star drive has been the most reliable drive I have ever encountered. (Two years and no repairs necessary) As far as a high speed tape system, I've heard the Alpha or Beta Drive was an excellent system that has worked with the Sol with no trouble. Anyone else have any ideas?

METEOROLOGICAL I/O

My main concern at this juncture, is availability of assistance in a project I am designing; specifically, the interfacing of digital meteorological instrumentation to my Sol. Will I be able to garner some BASIC, step-by-step, which button do-I press guidance in such a venture? I hope to reciprocate with areas in which I hold a modicum of expertise- I mention this only because I have found computer people "in the know" to be readily disdainful of those whose crime is a passion to learn and to know. I hope- since PT is defunct- the climate in PROTEUS will be more receptive to those whose goal is to stamp out personal ignorance.
I look forward to your response and a long membership in PROTEUS.

Phillip Isard, M.D.
3060 Bristol Road, Suite 167
Cornwells Heights, PA 19020

Dear Dr. Isard:

Your questions about guidance in interfacing a meteorological instrument to your Sol are hard to answer. If you want extensive help designing an interface and appropriate software drivers, you will find it hard to get much help unless you manage to locate someone who has similar interests. I will publish your desire for help in a classified ad, "Information Wanted". You might also try to locate a compute club in your area.

Regarding your perception that computer people "in the know" are disdainful of those whose crime is a passion to learn and to know, I must say that my perception is different. All of us have very limited free time to spend upon our hobbies. The

CONTINUED ON NEXT PAGE.

computer people I have had contact with (and I've been involved in computing since 1961) have been anxious to share wht they know, but they don't have the time to teach a total novice everything he needs to know. You'll have to do a lot of self-directed "digging it out" to learn the "how-to's" you want.

If hardware is your interest, a digital electronics course at a local college may give you the fundamentals you need to be able to avoid the "disdain." Imagine your reaction if a layman came up to you and asked for some basic, step-by-step instructions for the diagnosis and treatment of acute abdominal pain. You may be happy to give him advice, but that sort of question would just evoke a groan because of the scope.

Sincerely,
Stanley M. Sokolow, D.D.S.

Editor: I agree witn Stan that tne scope of the question will cause some dismay, but, I also agree with Phillip in that there is a need for some BASICS regarding the simple day to day operations of these tools. I guess you could equate it with asking a doctor how you could give yourself a blood pressure test. With the right equipment and a little assistance a lay person should be able to handle it. As far as tne other kinds of questions; deep in scope or not, lets get them asked and see what kind of answers we receive. Somebody might fool us and satisfy our needs with an answer we can understand.

ONE PORT AND TWO PRINTERS?

Dear Stan,

I have a Sol-2Ø with a SELECTRIC II hooked up to my parallel port. I would like to hook a faster printer to the Sol, and have the two printers hooded to the same port.

I was thinking of making up a box with the parallel cable on the input and hooking the printers to the two outputs with a switch to control the selected device with the DATA READY signal.

Do you or any of your readers know of a better idea?

John L. Gorman
21Ø Sprague Ave
So. Plainfield, N.J. Ø7Ø8Ø

SOL/NORTH STAR ELECTRIC PENCIL PROBLEMS.

Dear Stan,

I have a question and maybe you, or someone may have an answer. (I hope so) I have a Sol with a North Star single density drive and Electric Pencil I. The output is a Centronix 779 from the parallel port. The pencil will not double, triple, etc space. The page spacing does not work, nor does the page length. Everything else works fine. Michael Schrayer does not know what to do- he never heard of Centronics. Apparently the Pencil, or the Centronics Printer does not pick up the spacing information and I just don't know where to go.

Any ideas?

Roger Doran
Birch Point
West Bath, Maine Ø453Ø

CP/M TROUBLE WITH MICROPOLIS DRIVE

I am having trouble running CPM 11.41 and 11.31 tnru my Sol. I have a 48K system and a Micropolis Modl disk drive. If anvone close to my area can give me some help, I'd appreciate it.

Paul Kittle
P.O. Box 1285
Luma Linda, Ca 92345
714-796-13ØØ

WANTS TO PRINT AND SEE VIDEO AT SAME TIME

PROTEUS,

I have a problem which no one seems to be able to resolve. We use a "Qume" printer for "Word Wizard" with the PTC Sol. This system works beautifully, however, when I write a program using the PTDOS DBASIC I have a problem.

If I want to print out the program or run and print it I mut go into (open up) the "Sol" and the "QUME" to change the Baud rates to make them compatible, and even if I do this the program being run or listed is either on the printer only or on the VDM only. I would like it on both at the same time as well as not having to open them (the computer and printer) each time I go from WW to BASIC.

Is there anyway to get the driver for the printer from the WW for use with BASIC?

I would greatly appreciate any help and would be willing to pay a reasonable amount for it.

Jim Michaels, Marketing
Datarule Publishing Co, Inc
P.O. box 448
New Canaan, Ct. Ø684Ø
914-533-2263

[The WordWizard drivers can be copied from the WW System disk onto your BASIC system disk. They have type "IW" and should be changed to type "D" for use in BASIC. Your QUME driver is probably called "CUST".]

Stan

CP/M-NORTH STAR DIABLO DRIVER

Dear Mr. Solokow:

Please find $22.00 enclosed for Vol. 1 of the Solus New(1978) and for 1979 Proteus News. Since I already have numbers 1 and 3 of the latter, they need not be included.

In one of these, I noted some interest on the part of readers for a CPM/Sol Driver that would backspace for delete. If you haven't received adequate solutions for this yet, you might try the enclosed. It works fine for me. The relevant lines are 53 plus 84 through 101. The rest is evidently standard, but I include it for completeness. The elimination of the deleted characters from the screen makes working with CPM immensely easier.

In working with CPM I also might note that it is possible to run this operating system at 60K using the Northstar disks. It fits over the SOLOS routines without difficulty. However, most of the utilities have to be altered in order to prevent them from writing over the ROM. This is less simple, and I have one or two working, but not all.

For those SOL owners who have a Sol-Hytype interface, some may have difficulty in printing when their programs call for a skip over several lines. My interface, and I know of at least one other, drops the first character of the next line. While I believe this to be a bug in the hardware, by altering Line 40 in the driver printed in the user's manual to read: DCPAD EQU 1*5, that problem appears solved without loss of speed. Aside from this bug, the interface appears to be excellent and permits full use of all Diablo facilities. There is, however, a great deal of work necessary to develop the appropriate drivers since PT provided only a very limited driver in their manual. If there are a number who have done work here, an exchange of programs might be useful.

At a users meeting sometime back, Dr. Starkweather mentioned that he had adapted Pilot to the Northstar disk system and that he would make it available to the group. I am wondering whether anything has come of that.

In the two editions of PROTEUS that I have seen, you have done a good job of bringing various materials together. I hope that this can be kept up.

Sincerely yours,

Louis P Bucklin

Louis P. Bucklin

[I believe the problem with Hytype interface is that it doesn't allow enough time for the ribbon to lift after line feeds, or it doesn't realize that the ribbon has dropped. If you physically tie the ribbon lift mechanism in the up position with a tie-wrap or something similar, the dropped letter problem may go away with no other changes. – Stan]
It worked on my printer.

```
 1:          ;USER I/O ROUTINE AREA
 2:          ;
 3:          ;
 4:          ;EQUATES
 5: MSIZE    EQU     40      ;CHANGE THIS FOR DIFFERENT SIZE SYSTEMS
 6: BIAS     EQU     (MSIZE-16)*1024 ;CHANGES BIAS AND ORG FOR SYSTEM
 7: SSTAT    EQU     0F8H
 8: SDATA    EQU     0F9H
 9: IOBYT    EQU     3
10: USER     ORG     4400H+BIAS        ;NOTE: =BIOS+600H
11: SOUT     EQU     0C019H
12: AOUT     EQU     0C01CH
13: SINP     EQU     0C01FH
14: AINP     EQU     0C022H
15: LF       EQU     10
16: CR       EQU     13
17:          ;
18:          ;JUMP TABLE - CP/M JUMPS HERE FOR I/O
19:          ;JUMPS MUST REMAIN HERE, IN SAME ORDER
20: INIT     JMP     INITR    ;INITIALIZATION
21: CONST    JMP     CSTAT    ;CONSOLE STATUS CHECK
22: CONIN    JMP     CIN      ;CONSOLE INPUT
23: CONOUT   JMP     COUT     ;CONSOLE OUTPUT
24: LIST     JMP     LISOUT   ;LIST OUTPUT
25: PUNCH    JMP     PUNOUT   ;PUNCH OUTPUT
26: READER   JMP     READIN   ;READER INPUT
27:          ;
28:          ;BEGIN USER DRIVER ROUTINES HERE
29: INITR    XRA A ! STA IOBYT        ;STORE 0 AT LOCATION 3H
30:          ;INSERT YOUR OWN INIT HERE IF NEEDED BY YOUR SYSTEM
31:          ;INIT IS CALLED TWICE ON COLD BOOT AND ONCE EACH WARM BOOT
32:          ;USER CAN KEEP TRACK IF NEEDED WITH "BEEN HERE BEFORE BYTE"
33:          RET
34: CSTAT    LDA     CHAR     ;SEE IF STATUS ALREADY GOTTEN
35:          ORA     A
36:          JNZ     STAT2    ;IF GOTTEN SAY SO
37:          CALL    SINP     ;GET STATUS AND/OR CHARACTER
38:          STA     CHAR     ;HOLD ;CHARACTER IF THERE
39: STAT2    MVI     A,0      ;CHARACTER WAITING
40:          RZ
41:          CMA
42:          RET              ;RETURN OFFH IF CHARACTER READY
43:          ;
44: CIN      CALL CSTAT       ;CONSOLE INPUT ROUTINE
45:          ANA     A
46:          JZ      CIN      ;WAIT FOR CHARACTER
47:          LDA     CHAR
48:          PUSH    PSW
49:          SUB     A
50:          STA     CHAR
51:          POP     PSW
52:          ANI     127      ;STRIP PARITY
53:          STA LAST$C       ;SAVE AGAIN TO CHECK FOR DELETE
54:          RET
55: LAST$C           DB       0
56: CHAR     DB      0
57: FLAG     DB      0
58:          ;
59:          ;CONSOL OUTPUT ROUTINE IGNORES MORE THAN ONE
60:          ;CARRIAGE RETURN ON A LINE
61:          ;
62: COUT     MOV     A,C
63:          CPI     CR
64:          JNZ     LFCHK
65:          PUSH    H
66:          LXI     H,FLAG
67:          MOV     A,M      ;FLAG HAS NUMBER OF CR'S ON LINE
68:          INR     M        ;COUNT ONE MORE
69:          POP     H
70:          ANA     A
71:          JNZ     RESTORE ;IF NOT FIRST, IGNORE
72: LFCHK    CPI     LF
73:          JNZ     OUTC     ;CHECK FOR CR
74:          PUSH    H
75:          LXI     H,FLAG
76:          MOV     A,M
77:          MVI     M,0
78:          CPI     2
79:          JNC     ENDCHK
80:          MVI     C,LF
81:          CALL    OUTC
82:          MVI     C,CR
83: ENDCHK   POP     H
84: OUTC     PUSH B
85:          MOV A,C
```

CONTINUED ON NEXT PAGE.

```
         CPI 7FH          ;CHECK IF DELETE IS CURRENT CHARACTER
         JNZ C1
         POP B
         RET              ;DON'T PRINT THE DELETE
C1       LDA LAST$C       ;CHECK LAST CHARACTER FOR DELETE
         CPI 7FH
         JZ C2            ;JUMP TO BACKSPACE OUT ROUTINE
         MOV B,C          ;NOT DELETE, SEND OUT GOOD CHARACTER
         CALL SOUT        ;CHARACTER IN C IS OUT
         POP B
         JMP RESTORE
C2       MVI B, 5FH       ;SEND BACKSPACE TO CRT
         CALL SOUT
         POP     B
RESTORE  MOV     A,C
         RET
         ;
         ;LIST OUTPUT AND PUNCH ARE THE SAME
LISOUT   ;
PUNOUT   ;
         MOV     B,C
         CALL    0D003H  ;DIABLO DRIVER ENTRY POINT
         RET
READIN   ;
         MVI A,1
         CALL AINP
         ANI     127
         RET
```

HANGMAN / QUBIC FIX

Dear Tony,

In answer to the second item in Bob Stek's letter to Santa (PROTEUS NEWS, Sept/Oct 79), there is a simple cure. Qubic and Hangman both expect to find the SOLOS START address in the HL register pair when they begin execution.

The EX command in SOLOS provides this, the EXEC command in PTDOS does not. Obviously, neither does the Northstar DOS.

To make QUBIC flicker-free and executable as a PTDOS Image file, perform the following:

```
     ZIP 0 to clear memory; load QUBIC from tape.
     Do a reset to SOLOS. ENter 0720 :21 00 C0 C3 00 00
     EXecute BCB0 to get back to PTDOS
     IMAGE QUBIC,!4C0,0,725,720
```

Do the same thing for HANGMAN except ENter 0E20 :21 etc.
and IMAGE HANGMAN,!4C0,0,E25,E20

I don't have 8080 Chess, but I'll bet the solution is the same.

For those who do not have PTDOS, the syntax of the IMAGE command is: IMAGE FILENAME,!Blocksize,Beginning address, End address, Start or Execute address. The sequence above put an LXI H,0C000H instruction followed by a JMP 0000H onto the tail end of the program code. Save these new instructions along with the program and have your DOS Execute to the address of the LXI instruction.

Also, I am interested in the AccPac Programs, but would like to see them running before buying them. I'd like to visit with any user in Ohio, Kentucky, W. Virginia or Indiana who would share his experiences with these programs.

Ben

Ben C. Stapleton, Jr.
2430 Summit Street
Portsmouth, Ohio 45662

HELP NEEDED: CENTRONIX/DIABLO/WORDWIZARD

DEAR TONY;

I HAVE BEEN IN COMMUNICATION WITH STAN SOKOLOW ON SEVERAL OCCASIONS REGARDING WORD-WIZARD. I KNOW THAT YOU TOO HAVE CONSIDERABLE EXPERIENCE WITH THE "WIZARD" SO I WOULD LIKE YOUR OPINION ALSO. YOU WILL APPRECIATE THE REASON WHY I DELAY ORDERING THE THING WHEN I EXPLAIN OUR SYSTEMS AND NEEDS.

WE HAVE TWO SOL-SYS III'S AND A CENTRONIX-700 PRINTER. WE RUN BOTH PTDOS AND CP/M (LIFEBOAT VS.). THE CP/M GIVES US SEVERAL CHOICES IN WORD PROCESSORS; THIS LETTER IS THE PRODUCT OF DIGITALS ED AND TEX. WE HAVE SEEN WORD-STAR AND IT IS VERY, VERY GOOD, SUPERIOR IN SOME WAYS TO "W-W" BUT INFERIOR IN OTHERS.

OUR BIG PROBLEM HOWEVER IS IN INTERFACING. BECAUSE OF THE CONSIDERABLE DIFFERENCE IN PRICE , AND BECAUSE IT IS A PART OF A SYSTEM WRITTEN FOR OUR LARGER SYSTEM , I WOULD CHOOSE "W-W" IF I KNEW THAT I COULD USE THE CENTRONIX WITHOUT A LOT OF PROBLEMS. WE WILL BE BUYING A DIABLO OR EQUIVALENT THIS YEAR BUT BECAUSE 70 OR 80 PERCENT OF OUR COMMUNICATION IS IN-HOUSE AND DOES NOT REQUIRE LETTER-QUALITY AND BECAUSE WE MAKE AS MANY AS THIRTY COPIES OF OUR LETTERS WE NEED THE SPEED AND RUGGEDNESS OF THE DOT-MATRIX PRINTER.

WHAT DO YOU THINK TONY ? CAN I UTILIZE BOTH THE CENTRONIX AND A DIABLO, SELECTRIC OR SPIN-WRITER ON THE PRESENT VERSION OF WORD-WIZARD WITHOUT A GREAT DEAL OF TROUBLE? PLEASE LET ME KNOW IF YOU HAVE AN ANSWER; THIS FORMATTER IS A PAIN IN THE KNECK.

EARL J. DUNHAM
941 N. RUSSELL
LA HABRA CA 90631

ED: HELP, SOMEONE!

[Stan says: PTC Update 731075, Proteus item D25, describes spec's for custom drivers. You can modify the source for Sol3 supplied with WordWiz. But surely someone has a Centronix driver]

FORTH INTEREST GROUP

Dear FORTH Interestee:

F.I.G. is staffed entirely by volunteers. For this reason, our response to questions is often brief. Watch FORTH DIMENSIONS! All general information is distributed thru this medium.

F.I.G. publishes technical data and listings. We rely on companies and user libraries to supply running systems. We will publicize vendors in the newsletter.

In summary, we offer:

	US	OVERSEAS
FORTH DIMENSIONS 6-issue subscription	$5.00	$9.00
PRIMER, by Richard Stevens, KPNO. 230 pages. Introductory/exercises/ answers	$20.00	$26.00
using FORTH, by Forth, Inc. 200 pages. The best users manual	$20.00	$26.00
fig-FORTH Installation Manual. Model of the language and 220-word glossary	$10.00	$13.00
fig-FORTH Assembly Listings each (8080, 6800, PDP-11, 9900, PACE, 6502)	$10.00	$13.00

All items sent post-paid. Enclose check with order; no billing or purchase orders can be handled by our volunteer staff. Thank you for your interest.

FORTH INTEREST GROUP ····· P.O. Box 1105 ····· San Carlos, Ca. 94070

SOFTWARE

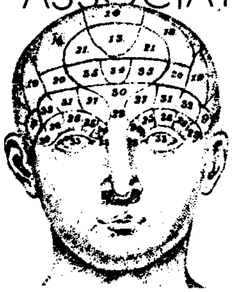
```
10 REM  ***     'PROGRAM PATCHDISK'     ***
20 '
30 '    THIS PROGRAM WILL ACCESS AN EDIT80 OR CPM SOURCE
40 '    FILE OF ANY LENGTH AND PATCH A RANDOMLY ACCESSED
50 '    RECORD ON A CHARACTER BY CHARACTER BASIS.  IT WAS
55 '    WRITTEN FOR 48K SOL-20 WITH D.D. NORTH STAR CP/M.
60 '
70 '    WITH MBASIC 5.0 OR LATER VERSIONS, VARIABLE
80 '    RECORD LENGTHS TO A MAXIMUM OF 128 MAY BE USED.
90 '    HENCE, A SEARCH MAY BE MADE FOR THE TARGET TEXT
100 '   USING THE MAXIMUM REC. LENGTH.  THEN THE LENGTH
110 '   MAY BE DIVIDED AND THE RECORD NUMBER MULTIPLIED
120 '   LIKEWISE TO NARROW IN ON THE TARGET CHARACTERS.
130 '   THE TARGET RECORD IS VERIFIED ON RETRIEVAL
140 '   FROM THE FILE AND ALSO AFTER PATCHES HAVE BEEN MADE.
150 '   THIS CODE MAY BE ADAPTED TO MBASIC 4.5 BY REMOVING RL
160 '   FROM LINE 560, AND MAKING IT A CONSTANT 128 IN
170 '   THE FIELD STATEMENT.
180 '
190 '   THIS PROGRAM IS PARTICULARLY USEFUL FOR MAKING SMALL
200 '   CHANGES TO EXTREMELY LONG ASSEMBLY LANGUAGE SOURCE
210 '   FILES WHICH OVERFLOW IN-MEMORY EDITORS, OR IN CASES
220 '   WHERE RECORD-BY-RECORD EDITORS ARE UNWIELDY.  FOR
230 '   BREVITY NO EOF TEST IS MADE.  LOF IS INVALID
240 '   BEYOND THE FIRST EXTENT.
250 '
260 REM         Copyright 1980, Daniel S. Hunt
270 '
280     CLEAR
290     DEFINT  J, E, R
300 '
310 REM *** CONSTANTS
320             ESC = 27
330             WIDTH 64
340 REM *** KEY VARIABLES
350 '           A$    :  INPUT RECORD BUFFER STRING
360 '           B$    :  OUTPUT RECORD BUFFER STRING
370 '           X$    :  TARGET CHARACTER

370 '           X$    :  TARGET CHARACTER
380 '           C$    :  MODIFIED CHARACTER
390 REM *** PROCEDURES
400 '           VERIFY INPUT RECORD
410 '           VERIFY OUTPUT RECORD
420 '           ASK USER IF O.K.
430 REM BEGIN...
440 PRINT CHR$(11)
450 PRINT "**** RANDOM ACCESS TEXT PATCH UTILITY ***"
460 PRINT "Beware! There is no test for end of file!"
470 INPUT "ENTER FILENAME <OR STOP>"; FI$
480 IF FI$ = "STOP"
     THEN
     PRINT "**** STOP ****" : END
490 INPUT "RECORD LENGTH"; RL
500     IF RL < 1 OR RL > 128 THEN PRINT "REDO ";:GOTO 490
510 ON ERROR GOTO 1120 :'TRAP FOR NON-FILE
520 OPEN "I", 1, FI$ : CLOSE
530 OPEN "R", #1, FI$, RL
540 FIELD #1, RL AS A$
550 PRINT "-----------------------"
560 INPUT "RECORD NO."; RN
570 IF RN < 1 THEN PRINT "**** ERROR ****": GOTO 560
580 PRINT "-----------------------"
590 '
600 B$ = ""
610 REM
620 GET #1, RN
630 GOSUB 880 :' VERIFY INPUT RECORD, ABORT IF INCORRECT
640 REM   EXAMINE AND/OR REPLACE EACH CHARACTER ONE BY ONE
650     PRINT CHR$(11)
660     PRINT "TYPE IN NEW CHAR, OR <ESCAPE> IF NO CHANGE ****"
670     FOR J = 1 TO LEN(A$)
680             PRINT
690             PRINT RL - J;" CHARS REMAINING:     ";
700             PRINT MID$(A$, J, 1);:PRINT "< OLD - NEW > ";
710             X$ = INPUT$(1)
720             IF ASC(X$) = ESC
                        THEN C$ = MID$(A$,J,1)
                        ELSE C$ = X$
730             PRINT C$;
740             B$ = B$ + C$
750     NEXT
760     GOSUB 960 :'VERIFY OUTPUT RECORD. ABORT IF BAD
770     LSET A$ = B$ :' ELSE PLACE IT IN BUFFER AND
780     PUT #1, RN : '  AND FILE IT
790     CLOSE 1
800 RESET :' UPDATE DIRECTORY FOR DISK REMOVAL
810 PRINT "**** DONE! ****"
820 CLEAR :' NEUTRALIZE ERROR SETS
830 END
840 '
850 REM ***********     SUBROUTINES     ***********
860 '
870 REM *** PROCEDURE: VERIFY INPUT RECORD
880 PRINT
890     FOR J = 1 TO LEN(A$)
900     PRINT MID$(A$,J,1);
910     NEXT
920     GOSUB 1040 :' FILE OR FORGET?
930     RETURN
940 'END;
950 '
960 REM ***  PROCEDURE: VERIFY OUTPUT RECORD
970 PRINT : PRINT : PRINT
980 FOR J = 1 TO LEN(B$)
990     PRINT MID$(B$,J,1);
1000 NEXT
1010 GOSUB 1040 :' FILE OR FORGET?
1020 RETURN :'END VERIFY
1030 '
1040 REM ***  PROCEDURE: ASK USER IF O.K.
1050 PRINT : PRINT
1060    PRINT "**** IS THIS CORRECT?";
1070    Y$ = INPUT$(1)
1080    PRINT
1090    IF Y$ <> "Y" THEN CLOSE :
             CLEAR :
             GOTO 10 :
            ELSE
             RETURN
1100 'END;
1110 '
1120 REM *** PROCEDURE : TRAP NON-EXISTENT FILE
1130    PRINT "**** FILE NOT FOUND. CHECK NAME ****"
1140 RESUME 10
```

DANIEL S. HUNT / PUBLISHING STRATEGIES / 16 Escapade Court / Newport Beach, CA 92663

DID YOU KNOW?

INTERNATIONAL **North Star** USERS ASSOCIATION



information sharing
breakthrough!

FOR ALL OF US

WITH

North Star
Disk Systems

REALIZE YOUR COMPUTERS FULL POTENTIAL!

1.A source of information for new as well as advanced users of North Star disk systems

2.A source of programs and utilites from a central software exchange library

3.A quarterly newsletter of practical techinques and application programs designed specifically for your North Star system

4.A liason between local users groups to promote the exchange of information usefull to all North Star users

5.An independent non-profit users represenitive providing feedback to North Star Computers inc.

INSUA MEMBERSHIP APPLICATION

( ) PLEASE BILL ME $15 YEARS DUES
( ) ENCLOSE PAYMENT & GET A FREE
EXTRA ISSUE OF THE NEWSLETTER

LAST NAME____________________
FIRST NAME____________________
STREET____________________
CITY____________________
STATE,ZIP____________________
PHONE____________________
USER GROUP?____________________
CONTACT____________________
GROUP SIZE____________________
ADDRESS____________________
ADDRESS____________________
PHONE____________________

SYSTEM SURVEY
COMPUTER: ( )HORIZON ( )OTHER
MEMORY: ( )<32K ( )>=32K
DISK: ( )ONE ( )TWO ( )THREE
DISK: ( )5" ( )8" ( )OTHER
DENSITY:( )SGL ( )DBL ( )QUAD
USE:( )BUSINESS ( )HOBBY ( )BOTH

MAIL MEMBERSHIP APPLICATION TO:
INTERNATIONAL NORTH STAR USERS
ASSOCIATION
131 HIGHLAND AVENUE
VACAVILLE, CA 95688

THE ONLY
INDEPENDENT
NEWSLETTER
BY AND FOR
North Star
SYSTEM USERS

JOIN
YOUR USERS
ASSOCIATION
TODAY

VECTOR GRAPHIC

Dear Proteus,
Will PROTEUS help a person with a Vector Grafic Box with an 8080A CPU board and Vectografic support boards. The I/O board is Tarbell as is the 32K ram boards. The teminal is a Soroc. The two floppy discs are eight inch Shugart 800/801 single density running the IBM format. The printer is a Diablo High Type One, System 75 Model V. The language is CBASIC2 running CP/M.

I hope it is possible that your club software can be used on my machine. I would like to join if it is possible to use your software.

Melvin E. White
748 Whitney
Avenal, Ca 93204

Editor: Got Me! Hope someone out there with more background can offer assistance.

==

SAM76 Language

Is there any way that our PROTEUS libraries can distribute SOLOS/CUTS Cassette or NorthStar non-CP/M versions of SAM76? This is a great language, but does not exist in a version usable by many Sol owners. Please look into possible conversions to Sol/North Star formats.

Daniel Hendricks
Irvine, Ca

Editor: I had a long talk with one of the originators of the SAM76 language at the Second West Coast Computer Faire (gosh I'm showing my age!) and I was impressed at the way he seemed to believe it was a fantastic language. I didn't even know what a language was at that time and had just bought my Sol by kit. He said there would be Sol versions out, but I never heard of them again, except for an occasional mention in news releases printed in the magazines. Does anyone know more about SAM76 and the Sol?

==

Binary ÷

Dear Stan,
I wanted to share with Proteus members a very simple routine for doing binary division. The method consists of successively adding the two's complement of the divisor to the dividend until the remaining divisor reaches zero or turns negative. The conventional method (used by INTEL in the PL/M library, for example) develops the 16 bits of the answer by a series of 16 test and shipts with an optianl addition to restore the working dividend.

By splitting the dividend into a high and low part, my method requires a maximum of 512 iterations compared to the conventional technique requirirng a fixed 16. At worst, my method is 7 times slower.

So much for the bad news. The good news is that my method is up to 4 time faster as long as the answer to the division is about 8Ø to 1ØØ or less. This is because the inner loop for my method contains only three instructions starting at DV, below.

It is precisely the low-answer case which dominates personal computing activity. For example, a routine which computed row-column addresses for a VDM driver would result in rows in the range Ø-15 and columns in the range Ø-63. Column address for tabbing on a printer result in numbers of the order 4Ø-132. An analysis of random operands shows 32K cases where an answer of 2 results and only 2 where 32K results. Thus, the random operatnd case is biased towards smaller answers.

The routine takes HL and divides it by DE (both positive). TLhe answer is returned in BC. The subroutine TWOSDE negates DE. The subroutine SUBDE subtracts DE from HL.

```
        DIVIDE      PUSH     H              ;SAVE DIVIDEND
                    CALL     TWOSDE         ;NEGATE DE
                    MOV      L,H            ;FORM HL/256
                    MVI      H,O
                    CALL     DV1            ;DIVIDE INTO HL/256
                    MOV      B,C            ;PARTIAL ANS * 256
                    MOV      A,L            ;PARTIAL REMAINDER
                    POP      H              ;RESTURE ORIG. DIVIDEND
                    MOV      H,A            ;PARTIAL ANSWER
        DV1         MVI      C,-1           ;INIT COUNT TO
        DV2         INR      C              ;O
                    DAD      D              ;SUBRTRACT UNTIL
                    JC       DV2            ;HL < O
                    CALL     SUBDE          ;RESTORE HL TO > O
                    RET
        TWOSDE      MOV      A,D            ;PERFORM 2'S COMP
                    CMA                     ;BY 1'S COMP
                    MOV      D,A            ;AND ADD 1
                    MOV      A,E
                    CMA
                    MOV      E,A
                    INX      D
                    RET

        SUBDE       MOV      A,L            ;SUBTRACT DE FROM HL
                    SUB      E
                    MOV      L,A
                    MOV      A,H
                    SBB      D
                    MOV      H,A
                    RET
```

This routine is an abstract of the PATB division routine.

Dennis Reinhardt

Software

This is a program to drive a Selectric IBM Model 73 I/O with a Sol Computer,..written by Raj Bhatia.

```
*       SUB ROUTINE FOR EXECUTING DELAY LOOP
DELAY   PUSH D                        C900    D5
        MVI  D..O                     C901    16 00
```

```
DEL     DCR      d              C903    15
        JNZ      DEL            C904    C2 03 C9
        POP  D                  C907    D1
        DCR.D                   C908    15
        JNZ  DELAY              C909    C2 00 C9
        RET                    C90C    C9

UPLW    DS                      C90F    FF (STATUS UC/LC)

DRIVE   LXI H/L                 C910    21 00 CA (LOAD ADDRESS OF TABLE)
        MOV A,B                 C913    78
        ADD L                   C914    85
        MOV L,A                 C915    6F
        MOV A,M                 C916    7E
        ANI 01                  C917    E6 01 (CHECK FOR CONTROL CHARACTER)

        JZ  CNTRL               C919    CA 50 C9 (IF YES GOTO CNTRL)

TYPE    MOV  A,M                C91C    7E
        ANI 80                  C91D    E6 80 (CK FOR UC)
        JNZ                    C91F    C2 3B C9 (IF UC GOTO UPCSE)
LCSE    LDA   UPCW              C922    3A 0F C9
        CPI   02                C925    FE 02
        JZ                      C927    CA 34 C9
        MVI   A...02            C92A    3E 02
ULC     MVI     D...DELAY       C92C    16 0A (DELAY UC/LC)
        STA  UPLW               C92E    32 0F C9
        CALL OUT                C931    CD E0 C9
AILC    MOV   A,M               C934    7E
        MVI   D...DELAY (3)     C935    16 17 (DELAY FOR UC/LC)
        CALL   OUT              C937    CD E0 C9
        RET                     C93A    C9

UPCSE   LDA   UPLW              C93B    3A 0F C9
        CPI  04                 C93E    FE 04
        JZ   AILC              .C940    CA 34 C9
        MVI     A...04          C943    3E 04
        JMP  ULC                C945    C3 2C C9
CNTRL   MOV   A,M               C950    7E
        CPI   X  '00'           C951    FE 00 (CK FOR UNDEFINED CHARACHTER)
        RZ                      C953    C8 (IF YES, RTN)
        CPI   X '08'            C954    FE 08 (CK FOR CONTROL CHR)
        JZ    CR                C956    CA 70 C9
        CPI   X '10'            C959    FE 10 (CK FOR BS)
        JZ    BS                C95E    CA A0 C9
        CPI   X '20'            C95E    FE 20 (CK FOR SPACE)
        JZ    SPACE             C960    CA 90 C9
        MVI    D...08           C963    16 08 (DELAY FOR TAB)

        CALL  OUT               C965    CD E0 C9
        RET                     C968    C9
        XX
CR      MVI   D...0A            C970    16 35 (DELAY FOR CR)
        CALL   OUT              C972    CD E0 C9
        MVI    D...0A           C975    16 FF ( ADDITIONAL DELAY FOR CR)
        CALL  DELAY             C977    CD 00 C9
        RET                     C97A    C9
        XX
SPACE   MVI   D...X 08'         C990    16 16 (DELAY SPACE)
        CALL  OUT               C992    CD E0 C9
        RET                     C995    C9
        XX
BS      CALL    SPACE           C9A0    CD 90 C9
        MVI  D...X'0A'          C9A3    16 30  (BS DELAY)
        CALL   DELAY            C9A5    CD 00 C9 (ADD'L DELAY FR BS)
        RET                     C9A8    C9
        XX
```

CONTINUED ON NEXT PAGE.

SELECTRIC DRIVER...CONTINUED FROM PAGE 15.

```
        * MAIN PROGRAM. ENTER HERE FROM SOLUS *
SOUT        PUSH PSW                  C9B0    F5 (SAVE PSW)
            PUSH  D..E                C9B1    D5 (SAVE REG D&E)
            PUSH     H...L            C9B2    E5 (SAVE REG H & L)
            MOV  A,B                  C9B2    78
            STA   ADR1                C9B4    32 CF C9 (SAVE ACC)
            CPI  X '80'               C9B7    FE 80 (CK FOR LEGAL CHR)
            JM    START1              C9B9    FA C9 C9 (IF YES GOTO START1)

            MVI    A,0                C9BC    3E 00
            LXI    H,ADD 1            C9BE    21 CF C9
            MOV    B,M                C9C1    46
            CALL   AOUT               C9C2    CD 1C C0
            POP  H                    C9C5    E1
            POP  D                    C9C6    D1
            POP   PSW                 C9C7    F1
            RET                       C9C8    C9
            CALL   DRIVE              C9C9    CD 10 C9
            JMP      RET              C9CC    C3 BC C9
            DS                        C9CF     (SAVE ACCUMULATOR)
            XX
 OUT(MAIN)
            OUT FD                    C9E0    D3 FD
            CALL DELAY                C9E2    CD 00 C9
 ENERG      MVI  A...0                C9E5    3E 00
            OUT   FD                  C9E7    D3 FD
            MVI    DE DELAY           C9E9    16 08
            CALL   DELAY              C9EB    CD 00 C9
            RET                       C9EE    C9
 XX END, TO LOOK UP TABLE
```

GREETINGS FROM MICROCOMPUTER RESOURCES!

I would like to **comment** on several things that I noticed in Volume 2, #5 of PROTEUS NEWS. Joe Maguire brought to light what he though was a 'bug' in PTDOS. The first involved the operation of the "READ" command. He found that when he used a READ to load memory with an Image file, he found gaps in the code, and some wrong bytes. That is the way PTDOS is supposed to work. The READ command will transfer all the bytes in a file. The Image command files byte counts and the load address of your program when it stores it in the file. This allows overlaying of data. The read command brings these in as well as the operating code. That is not a bug, it is a feature. To get around that problem, we have donated two programs that will make things a bunch easier. First is LOAD, which will do an offset load of an image file. You specify the amount of the offset when you invoke the utility, IE: LOAD BASIC,1000 to move it forward 1000 hex bytes. LOAD BASIC,E000 will move it backwards 2000 hex bytes. Get out your TI programmer to get offset loads the easy way. If you want to handle an Image file, you have to take care of the details.

Another comment per the Programming Quickie provided by Lewis Moseley, Jr. We are even lazier then Lewis when it comes to repetive work in SOLOS. If we need to use the SAVE command over and over, we Type: Cursor Up, and Line Feed. The use of Line Feed is non-destructive, and has the same effect in far fewer keystrokes. A Line Feed on a command line has the same effect as a Return, but can be done without erasing the rest of the line - it is non-**destructive**.

One last comment. MicroComputer Systems, Inc on South Dale Mabry Hwy no longer has any factory trained technicians. Jim Holloway (one of the best PT techs south of the Mason Dixon line) left MSI for greener pastures. That leaves us as the PT repair shop in Tampa. Warren Startup and myself are both factory trained, and are happy to help people with problems with their equipment. We are now the only people in the central area of the state that are factory trained. We are probably the only people in the state who know the SOL and HELIOS. Our rates are reasonable, and we can make service calls (for software or hardware!)

One comment about memory boards we have used and are now using. The Measurements Systems and Controls boards dynamic board works well in a SOL-North Star system. 48K is the upper limit, but it is good, low cost memory. Their board will not work with the HELIOS despite the mods they suggest in their manual. The Helios II controller can write to their board, but cannot DMA out of it. We are working on the problem with them, and if we come to a conclusion, we will tell you about it. Currently, we cannot recommend it for HELIOS II owners. The M-XVI from California Computer Systems, a static 16K board, seems to work nicely with the HELIOS, however. It includes bank select, and a number of other features. It seems to be very well executed. We have several under test, and all things seem to be positive so far. The EXTENSYS RM-650 does indeed work with the HELIOS II-Very well. The only board we found that is close is the PT NK-RA. Too bad they went belly up. It seems like making a board that works with the HELIOS II is fatal. Let us pray for California Computer Systems....

We are a dealer for The SOURCE. If any Proteus people would like to subscribe, we can help in several ways. We have a modem control package for the DC Hayes S-100 board that allows a SOL-North Star system to communicate with the outside world. It allows file transfers, and so forth. If we sell the board, we will include the driving software at no cost. Same thing applies for a Helios II owner. That driver is not up yet, but is in the works. It should not take too long to make it run. Also we will discount the SOURCE to PROTEUS members. We can supply it for $80.00 (plus postage.) The regular cost is $100, so this is a fair discount.

We will be working on software to connect the new N* hard disk to PTDOS (running with a Helios II on a SOL.) We know that there are lots of people looking for more storage, and it is an economical way to go...$5000 for 18 Meg. We intend to implement it as drive 0, with a second reserved as 1. The Helios II will think it is 2 and 3 (with no hardware changes...) The Floppy would be only used for backup and short duration transients. The hard disk plugs into the parallel port, so it does not fill your limited space SOL. Incidentally, we also intend to put PTDOS on the Horizon, making software you generate now transport able to other gear.

Our catalog is on disk. It's in PENCIL form, so it has no cr/lf's stuffed into it. Don't try to view it in EDIT.

That is all for now. Keep up the good words (hopefully in byte reversed order....)

Bill Blomgren, VP

3000 Medical Park Drive, Suite 107-108

Tampa, Fl 33612

(813) 977-5940

NEWS FROM PROTEUS
by Stan Sokolow

I HAVE THE SOURCE CODE!

At long last, I have actually gotten the source code disks for PTC's software, as we have been discussing for 8 months now. There are a few missing programs that PTC is going to look for again. Among these are: ALS-8, Level 1 Business BASIC, GAMEPAC2, and a number of PTDOS commands.

A few of the ones I originally was told were available actually are not available because PTC doesn't own the source code, only the right to distribute object code. People who ordered these will be given the option of a refund or credit toward another source program, unless I can track down the original authors to work out a deal with them. Among these are: DEBUG, TREK-80, GAMEPAK1, EDT3, Cassette EDIT, FOCAL.

I plan to try to arrange some type of deal with the authors of other programs which run under PTDOS. If you have written a program which you feel is good enough to sell (such as a business program or a scientific application), please contact me. Proteus can handle all of the hassles and you can simply collect the royalties, or you can do it all yourself.

We do have a surprise program: SOLED work-in-progress. Someone may want to work on completing it.

I've run an assembly listing of the PTDOS resident code—it is 200 pages long! With all of the command files, PTDOS will probably be 800 to 1000 pages of assembly listing altogether. Considering the amount of time it must have taken to produce that much debugged code, our deal is quite a bargain.

After I have received the missing files and examined them carefully, I will announce in the newsletter the availability of the program sets I outlined in the "catalog" long ago. Those who have already sent advance payment will be contacted individually by mail.

HELIOS UPGRADE PROGRESS REPORT

I haven't had much time or incentive to look further into a replacement for the Helios controller, but now that PTDOS source code is in hand, I want to get going on that project again. I know the Micromation Doubler works, but I want something better.

TONY SEVERA REPORTS FOUL-UP SHOULD BE FIXED

Tony Severa, the editor, has reported to me that the clerical mix-up which delayed or lost some members orders placed with him should be all corrected by now. Anyone who hasn't received something they ordered from Tony should contact Tony if they haven't done so within the past few months. Anyone who feels that he hasn't been treated correctly should contact me, Stan Sokolow, at the Proteus address and I'll investigate or expedite to the best of my ability.

PLEASE ORDER FROM THE RIGHT ADDRESS

My secretary has requested that I ask members to be careful when they address correspondence to Proteus. BE SURE THAT YOU SEND YOUR CHECK AND ORDER TO THE RIGHT PERSON. Tony Severa is handling the editorial contributions for publication in the newsletter (that is, articles and letters) and the reproduction of PTC documents listed in the "Proteus catalog" in prior issues (items having the prefix "D" on their Proteus item number, such as "D23"). Make payment directly to Tony for the D-items. His address is 131 Highland Avenue, Vacaville, CA 95688.

Lewis Moseley, Jr., is handling the software library on cassettes. This includes software that runs under CP/M as well as on the PTC system software. See the back issues for more details. Lewis's address is Lewis Moseley, Jr., 2576 Glendale Ct NE, Conyers, GA 30208. Make the checks payable to him.

I (Stan Sokolow) handle the subscriptions to the newsletter, the proprietary programs (P-items in the catalog, such as the Basic Computer Group's software packages), the source code purchase, and other oddball things from time to time. Place orders and send payment to "Proteus", Attn: Stan Sokolow, 1690 Woodside Road, Redwood City, CA 94061.

RUMOR QUASHED

A few members have pointed out a remark in an advertisement in the "Computer Shopper" tabloid. A classified ad place by one Brian Gildea, 1246 Smallwood Dr, Columbus, OH 43220, purported that "Sol chassis are being produced again with the computer to follow. I will guarantee [items being sold] for 6 months and they should be in production by then."

I have checked with the ultimate source: an officer of PTC who would be the one to negotiate and sign any such licensing or sale of the manufacturing rights. He says that this rumor has no basis.

Mr. Gildea may be thinking of a so-called "Swedish Sol", which is a nickname for a new computer that is a cross between the Sol's design and a TRS-80 design. It's actual name is probably going to be "The Expander". It is designed by the same engineer who made the Sol, Lee Felsenstein, so naturally there are similarities. But still, it's not the Sol.

ANYONE WANT ANOTHER SOL OR HELIOS?

I have been contacted by a number of people who want to sell extra machines. Some have only been used long enough to check that they were working properly. Contact me, Stan Sokolow, for details. Prices vary, but will be around $900 for an empty Sol, $1500-2000 for a Helios, $400 for 64KRA Ram module.

THIS IS THIRD ISSUE OF 1980

We have received a few requests to send what the writers thought were missing issues. As Tony Severa mentioned in the last issue, he got behind schedule after the West Coast Computer Faire. The second issue had no number, but said it was the "April/May 1980" issue. This is the third issue. It actually sat on my desk a few weeks longer than it should while I wrote my page. I wanted to report good news about the source code, and it seemed imminent for a long time. Rest assured that you will receive 6 issues in 1980, as promised. Tony has delivered to me the next issue's camera-ready copy, so expect it to follow this one as soon as I get the source code organized. The source code prices, etc., will be repeated on my page in that issue.

PRINTERS UNDER $1000 ANYONE?

I'm in the market for one of the under-$1000 printers that have been coming out over the past year. Specifically, I want one that prints both 80 and 132 columns on lettersize continuous forms (9.5" wide paper including the perforated margins), with adjustable tractor feed, and is capable of continuous duty. There are a number of these, such as the Base2, the IDS 440 Paper Tiger, the Heath/Zenith, the Okidata Microline 80, etc. I have heard that at least one of these is unreliable when used for long printouts. If anyone has experience with any of this type of printers, please let me know which one you feel is good or bad. If you own one, please send me a sample of the printing (bidirectional mode if available). I'll report the comments in a future issue.

PUBLICATION OF MEMBERSHIP ROSTER

From time to time, I have considered publishing a roster of Proteus members, but have never done it because of the risk that it could be used as a shopping list for burglars. Yet members occassionally ask to have names of others in their area.

I am considering making names available, but not publishing them in the newsletter en-masse. The roster would take up too much space and wouldn't interest everyone. Now that our mailing list is on a mailing-list management program, I can selectively produce lists. If you would like your name EXCLUDED from any published list, please write to let me know (Proteus, Redwood City address).

BACK ISSUES

We still have many sets of 1979 Proteus News back issues for sale at $15 per set, including postage and handling. The 1980 subscription rate is $18 for 6 issues, bimonthly approximately.

The 1978 and earlier issues are out of print. Someday I may publish a collection of the best of the back issues articles.

What's New?

The New and Better H - 4

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

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

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

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

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

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

QUIKSORT same as above but with modifications suggested by Knuth.

HEAPSORT from Knuth

SHELSORT from Knuth

SORT is another sort suggested by Knuth which seems to approach the speed of the quiksort but may not have the diadvantages(when the file is in order, for example) Knuth rates the mathematical evaluation of this sort at 5Ø points, his maximum.

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

FIND+ was originally inspired by the IEEE micro-mouse contest. This was the best program I came up with, and it naturally uses a recursive function. Note that it will require lots of memory to run! SET your BUffer = 9ØØØh and use the MBASIC on this disk.

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

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

ACCOUNTS is a sample list of accounts used witn BANKERS.

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

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

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

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

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

gameDOC is some documentation on the game programs that follow:

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

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

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

QUME is a driver and source code which allows the QUME SPRINT 5 printer to operate at 12ØØ baud. Has build in handshaking.

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

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

```
 NAME      TYPE  SIZE     FIND+       Ø5      8     ROBOTS:C    T       4
-+-+-+-+-+-+-+-+-+-+-     FORMAT      .       8     SER.RNDM    Ø5      4
!MESSAGE    M     16      FSMAP       ØØ      1     SERIAL      Ø5      4
**H-4DOC    .     16      HEAPSORT    Ø5      4     SET         .       4
ACCOUNTS    Ø8     4      HOWDY       .      2Ø     SHELSORT    Ø5      4
ACCTS.2     Ø7     4      KWIKSORT    Ø5      4     SORT        Ø5      4
ADD.WK      Ø5    12      LUNAR.1     Ø5     12     SP:C        T       4
ANYMO       Ø8    28      MAKDRIVR    $       4     SP:D        T      12
BANKERS     Ø6    12      MAZE        Ø5      4     SPB         D       4
BIORYM      Ø5     8      MBASIC      I.     64     SPB:S       S       4
BOOTLOAD    IØØ    3      MMIND       Ø5     16     SPO         D       4
CHASE       IG     8      NEXTID      ØØ      1     SPO:S       S       4
CONTENT1    .      4      NUMSTR      Ø5     4Ø     SYSGLOBL    ØØ      3
COPY        IS     4      PATCH1.5    IS      4     TENSOR:C    T       4
CREATFIL    Ø6     4      PERMTEST    Ø5      4     TENSORTS    Ø5     28
DIRECTRY    ØØ    16      PERSONAL    Ø6     12     TT          Ø8      4
EDIT+H19    S     12      PRIMES      Ø5      4     WARRANTY    .       4
ELIZA       Ø5    2Ø      QUIKSORT    Ø5      4     WK.DOC      .      28
EXPENSES    Ø8    2Ø      QUME        D       4     build       $       4
FACTOR      Ø5     4      QUME:S      A      12     mes.s       .       8
FILES       IC     8      README      T       4     mes.temp    .       4
FIND        Ø5     8      RETYPE      IC      4     message     I.      4
                                                    text        T       4
```

AUTHOR EXPLAINS NUMSTR AND ADD.WK ON H-4

DEAR BRO.

A NUMBER OF PEOPLE HAVE ASKED HOW WE USE THE SORT AND DATA HANDLER ROUTINES I SUBMITTED TO HLIB SOME TIME AGO. THE PRESENT PROGRAMS, NUMSTR AND ADD.WK, WORKING TOGETHER, ARE ONE OF THE PRACTICAL APPLICATIONS WE USE THAT INCORPORATE THOSE AND OTHER ROUTINES IN OUR DAILY BUSINESS.

NUMSTR IS ADMITEDLY LONG, BUT IT DOES A DIFFICULT JOB FOR US AND BECAUSE IT IS LOOSELY STRUCTURED IS EASILY MODIFIED AND HELPS US TO GENERATE SOME OF THE MANY REPORTS REQUIRED IN OUR SUPERVISION OF THIRTY-ONE RETAIL STORES FOR LIFE UNIFORM CO. IT IS EASILY IMPLEMENTED BY RELATIVELY UNSOPHISTICATED HELP BECAUSE OF THE LEAD-BY-THE-HAND, STRUCTURED WAY IT IS WRITTEN.

MUCH WILL SEEM UNECCESARY AND REDUNDANT TO THE OLD MASTER PROGRAMMERS; WE FEEL IT WORTH WHILE IN ITS SIMPLICITY BECAUSE WE SELDOM HAVE TO EXPLAIN MUCH TO THOSE WHO SPEND THERE TIME KEYING IN THE DATA AS A PART OF THEIR DAYS WORK. IT WOULD SEEM BETTER BREAKING IT UP INTO SMALLER PROGRAMS BUT IN PRACTICE THAT IS WHERE MOST THE ERRORS ARE MADE. SO IT IS THE WAY IT IS.

WHILE MUCH IS PECULIAR TO OUR SPECIFIC NEEDS, THE RATIONALE BEHIND THE VARIOUS MODULES ARE APPLICABLE IN MANY PRACTICAL AND COMMON NEEDS IN NEARLY ANY SMALL BUSINESS AND PROBABLY IN MUCH THAT ONE WOULD LIKE TO USE HIS COMPUTER FOR AROUND THE HOUSE. OUR DATA-FILES ARE QUITE LARGE, BUT THE SHELL-METZNER SORT ROUTINES HANDLE THEM WITHOUT UNDUE WAITS. I AM INCLUDING ONE OF OUR DATA-FILES SO THAT THE USER CAN RUN THE VARIOUS MODES AND SEE HOW EASILY AND QUICKLY GOOD LOOKING REPORTS ARE GENERATED BY NUMSTR. THE MENUS NEED LITTLE EXPLANATION, SIMPLY TYPE IN THE NAME OF THE DATA-FILE (THE ONE I INCLUDED - ANYMO - IS AN ACTUAL WORKING FILE; I HAVE ONLY CHANGED THE STORE NAMES FOR OBVIOUS REASONS) WHEN REQUESTED. START WITH THE FOUR-WEEK, NO HARD-COPY ROUTINE AND RANDOMLY INPUT RECORD NUMBERS BETWEEN ONE AND THIRTY WHEN ASKED TO DO SO. IT IS A GOOD EXAMPLE OF THE POWER OF RANDOM ACCESS FILES IN PTDOS.

THE TWO PROGRAMS WORK TOGETHER. AFTER THE DATA-FILE IS CREATED IN NUMSTR, SUBSEQUENT WEEKS DATA IS ADDED TO THE FILES IN ADD.WK. IT RUNS VERY FAST AND THE WEEKLY FIGURES FOR THIRTY STORES ARE ADDED TO THE FILES IN MINUTES. THE HARD-COPY AND VDM REPORTS ARE ALL IN NUMSTR. I HAD TO INCLUDE THE DATA-FILE ON THE SAME DISK AS THE PROGRAMS, SO IT WILL BE NECESSARY TO EITHER 'GET' IT TO YOUR UNIT/1 OR CHANGE THE PROGRAMS AT THE DATA-FILE SELECTOR MODULE AND DELETE LINE 7100 WHICH ADDS THE "/1". THE WHOLE THING REALLY WORKS BEST WITH BOTH COMMAND AND DATA-DISK RUNNING TOGETHER. WE BELIEVE THAT IT SAVES UNDO WEAR AND TEAR ON UNIT-ZERO'S HEAD TO USE THE DATA-DISK IN /1.

IF YOU WISH TO START OUT BY CREATING YOUR OWN DATA-BASE OF SOME SORT, TRY RUNNING NUMSTR AS IS UNTIL YOU GET THE HANG OF IT WITH A COUPLE OF SMALL FILES. SIMPLY SELECT THE PRINT ROUTINE FROM THE FIRST MENU. RESPOND TO THE PROMPTS WITH FOUR OR FIVE DIGIT INTEGERS WHEN IT ASKS FOR WEEKLY SALES. OUR STORE NUMBERS ARE TWO OR THREE NUMBER DIGITS AND THE STORE NAME CAN BE UP TO TWENTY CHARACTERS IN LENGTH. THESE COULD OBVIOUSLY BE ACCOUNT NAME AND NUMBERS IF YOU WISH TO ACTUALLY USE THE THING IN SOME PRACTICAL WAY.

UNLESS YOU HAVE A 700 SERIES CENTRONIX PRINTER, YOU WILL HAVE TO CHANGE OR ELIMINATE THE CONTROL CHARACTERS THAT ARE NOT APPLICABLE TO YOUR PRINTER. CONTROL-L IS PAGE ADVANCE, CONTROL-K IS VERTICAL TAB, THE VERTICAL BAR GENERATES DOUBLE WIDTH LINES ON OUR CENTRONIX-700. THOSE SYMBOLS WHERE ONLY USED AS PRINTER CONTROL WILL BE FOUND AT THE FOLLOWING LINES :

4352 - 4360 - 4870 - 4910 - 5225 - 5310 - 5700 - 5730 - 5760
6170 - 10140 - 10496 - 10605 - 11010 - 11450 - 11820 - 11995

VERTICAL BARS WILL BE FOUND AT THE END OF LINES:

4170 - 4680 - 5180 - 5690 - 9840 - 10220 - 10520 - 11150
11520 - 11920

IT PROBABLY WILL DO NO HARM TO LEAVE THEM THERE EXCEPT THEY WILL SEEM NON-SENSICAL IF THEY SIMPLY PRINT.

AS TO WHY THERE ARE SO MANY OPTIONS; ONE MUST DEAL WITH THE PROBLEM OF FOUR AND FIVE WEEK MONTHS, FISCALLY SPEAKING, WE HAVE DONE SO IN THE SIMPLEST MANNER. THE SORT OPTIONS HELP US IN TWO WAYS; WE USE A NUMBER OF INCENTIVE PLANS TO GENERATE SOME SALES ENTHUSIASM AND COMPETITION AMONG STORES. BEING ABLE TO SEE WHERE THEY STAND RELATIVE TO OTHERS IS A BIG HELP. MORE IMPORTANTLY WE CAN SEE AT A GLANCE IF A STORE IS HAVING PROBLEMS AS IT MOVES UP OR DOWN THE LIST. WE HAVE LEARNED TO SPOT A PROBLEM BEFORE IT GETS OUT OF HAND BY PAYING ATTENTION TO THESE REPORTS AND TO OTHERS WE CREATE THAT DEAL WITH OTHER PARAMETERS OF STORE OPERATION. ONE COULD EASILY USE THE SAME TYPE OF GOAL SETTING AND PERFORMANCE REPORTING IN SUCH THINGS AS PERSONAL EXPENDITURES.

THE WORD "BUDGET" MAY SEEM STRANGE, IT SIMPLY REFERS TO OUR PRESET GOALS FOR STORE PERFORMANCE BASED ON PAST HISTORY AND THE PREVAILING ECONOMIC SITUATION IN STORE AREAS.

THE OPTIONAL VDM-ONLY DISPLAY IS VERY USEFUL TO US FOR FAST LOOK-UP AND FOR CHECKING BEFORE WE START THE HARD-COPY ROUTINE. IT IS FUN TO SIMPLY INPUT RECORD NUMBERS AT RANDOM AND WATCH THE THE WHOLE REPORT JUMP UP ON THE SCREEN. THAT BRINGS UP A VERY IMPORTANT POINT. WE MAINTAIN THE SAME RECORD NUMBER FOR EACH STORE IN ALL OUR DATA-FILES. THAT WAY DIFFERENT PROGRAMS MAY EASILY COLLATE DATA FROM THE SAME STORE FROM ANY NUMBER OF FILES IN THE SIMPLEST MANNER. WE HAVE USED THIS SYSTEM SO LONG THAT WE ACTUALLY CAN RECALL EACH STORES RECORD NUMBER AT WILL. IT REALLY MAKES AN EASY TASK OF RECALLING THE RECORDS OF ANY STORE WITH OUT THE NEED FOR COMPLICATED SEARCH ROUTINES. TRY IT.

TO MAKE THE REPORTS SENSIBLE IT IS BEST TO INPUT ZEROS FOR FUTURE WEEKS WHEN INPUTTIN DATA. BUT I'M SURE THAT UNIQUE NEEDS WILL DICTATE CHANGES IN THE REPORT THEMSELVES. TRY RUNNING THE PROGRAMS AS IS WITH THE INCLUDED DATA-FILE AND THE IDEAS FOR NEW FORMATS WILL COME TO YOU.

THE MODULE CALLED "REGIONAL SALES SUMMARY" TOTALS DATA FROM ALL THE STORES INTO ONE REGIONAL REPORT. A VERY SIMPLE REPORT, IT GIVES US A LOOK AT HOW WE ARE DOING AS A REGION. WE HAVE OUR "BUDGETS" JUST LIKE THE INDIVIDUAL STORES.

ABOUT "ADD.WK". OUR ROUTINE IS THIS: AFTER CREATING A NEW DATA-FILE AND IMPUTING THE FIRST WEEKS DATA, SUBSEQUENT WEEKS ARE ADDED TO THE FILE BY RUNNING ADD.WK. IT IS VERY FAST, ONLY MINUTES ARE REQUIRED TO UPDATE AN ENTIRE 30 STORE FILE. IT RUNS ALMOST AUTOMATICALLY AFTER MAKING THE PROPER SELECTION, SIMPLY FEED IT NUMBERS AND THE DATA-FILE IS READY FOR NUMSTR IF ONE WANTS REPORTS FOR THAT WEEK.

I MUST DEFEND THE WASTE OF A PAGE OF PAPER AS YOU WILL SEE HAPPENS BECAUSE OF THE OPTION TO SORT OR NOT TO SORT AND IF SO TO BE SURE THE OPERATOR HAS REPOSITIONED PAPER IN THE PRINTER. WE COULD HAVE AVOIDED THIS BY CLOSING AND REOPENING THE PRINTER-DRIVER, BUT MORE PROBLEMS WERE ENCOUNTERED AND MORE TIME WASTED THAN THE SINGLE SHEET OF PAPER WAS WORTH.

THE STORE NUMBERS 60 THRU 250 ARE REAL TO US AND COULD BE CHANGES TO ANY CODE DESIRED. THE STORES ALSO HAVE NAMES (NS) WHICH COULD BE ACCOUNT NAMES. THEY USUALLY REFER TO CITY NAMES FOR US AND I HAVE CHANGED THEM TO KEEP OUT OF TROUBLE WITH OUR BOSSES. THE THIRD ENTRY "B" IS THE BUDGET WHICH I EXPLAINED BEFORE. WE ARE A RATHER STABLE ORGANIZATION AND SELDOM HAVE TO DELETE STORES FROM OUR FILES. ONE COULD DO SO BY SIMPLY OVERWRITING THEM WITH DATA FROM THE LAST ENTRY. BUT THE USE OF SAME RECORD NUMBERS WOULD HAVE TO BE DEALT WITH.

THERE IS LITTLE MORE TO ADD. IT IS ALL QUITE SIMPLE, AND EASY TO USE. WE OFFER IT ONLY AS EXAMPLE OF COURSE. WE DO NOT PRETEND THAT IT IS THE GREATEST THING EVER, BUT IT WORKS WELL FOR US. THE WHIZ-PROGRAMMERS WILL FIND MUCH TO FAULT AND I'M SURE MANY SUGGESTIONS TO IMPROVE IT WILL BE OFFERED. I HOPE SO. WE WILL NOT RESENT ANY CRITISISM, IN FACT WE ENCOURAGE IT. IF ANYONE FINDS MERIT IN IT WE WILL BE GLAD TO SHARE OTHER PROGRAMS WE USE IN OUR DAILY JOBS. ONE WAY OR THE OTHER WE WELCOME YOUR RESPONSE. I HAVE PERSONALLY FOUND THE SMALL COMPUTER THE MOST ENJOYABLE CHALLENGE YET. THERE IS SO MUCH TO LEARN AND SO MUCH FUN IN DOING SO THAT ALL OTHER AVOCATIONS SEEM TRIVIAL.

EARL DUNHAM
941 N, RUSSELL
LA HABRA, CA
90631

EDITOR: CONTINUED

SOLOS PERSONALITY MODULE RELOCATED TO END OF MEMORY

Lee Felsenstein, co-designer of the Sol, has produced a batch of personality modules for 2708 EPROMs with Solos relocated to F000 instead of C000 where it usually is. A simple modification to the Sol personality module socket allows the system to automatically adjust itself to power-up into the proper location (C000 or F000) depending upon which module is inserted. The entire 4K block of memory space used by the Sol is thus relocated automatically when the module is inserted. You can run the present software dependent upon Solos at C000, and also you can swap to the other module when you want to run CP/M or some other program that needs more than 48K of contiguous space. (Many of the big software packages, such as the Pascal's and other compilers, need lots of memory.) If you want the new module, contact Lee at the address below. Quantity one price is $129, but dealer discounts are available.

Lee Felsenstein
Golemics, Inc.
2608 8th Street
Berkeley, CA 94710

Telephone
415/848-2693

NEED A DELAY LINE OR nKRA?

John Edwards, 408 13th Street, #545, Oakland, CA 94612, 415/462-3394, has a few extra delay line devices ($19) for the nKRA memory boards (16KRA-1, 32KRA-1, 48KRA, and 64KRA). He may also have some extra PROMs for these board. He also is selling a 64KRA board in working condition, factory assembled and tested, $400. He does NOT have any delay lines for the original 16KRA board.

COOL OFF YOUR HOT SOL

We will install an extra fan for you if you live in the Houston area; carry-in service only. $75. F.D. Systems, Inc., 9000 SW Freeway, Suite 115, Houston, TX 77074, 713/995-5455. We also have some extra Sol System III's and separate Sol's for sale.

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, USA.

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

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

In this issue:

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

18104

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

Volume 3, #4

July/August 80
Sept / October

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

Single Issue $3.00 US

output

by Tony Severa

We start this issue with a large article written by our friend Fr. McGanee entitled, "Understanding your Sol..I/O Routines." I wish we had more of this type to be able to offer our readers! Everything Fr. McGahee produces is always of an excellent nature and is fully documented. I have heard his name and reputation many times.

I would like to welcome Al Olson of Grass Valley, Calif. who has kindly offered his energy and expertise in helping me with the Helios Library. I was able to visit with him up in the mountains that separate Nevada from California and it was just beautiful! It was strange to find such an experienced and dedicated user in the mountains. Anyway, Al has kindly agreed to go through the software that I receive and see to it that it works (to the best of his ability) and help me put together the disks.

I have also placed in this issue the first of several articles I am writing regarding business and micro-computers. This issue deals with the basics of micro-computers and the history of the micro as seen through my eyes. Hope you find it enjoyable and informative.

Last month I saw the final demise of the first computer store that I bought my first Sol from. They used to be called The Byte Shop of Walnut Creek, Calif. and recently changed their name to MicroSun Computers of Walnut Creek. Their demise was a long time in comming. They were one of those stores that decided to go with the Sol as much as possible when Processor Technology was making all sorts of crazy demands on its dealers.

They had put all their energy and money into supporting the Sol until P.T. closed their doors. At that time they had 25 businesses set up with the Sol/Helios system and were supporting those companies with a 24hr maintenance contract. They had several extra System III's but as time went on, slowly had to sell their back-up systems for operating capitol. Being burned by P.T. and experiencing similiar problems with North Star Computers they decided to build their own system on a modular basis using the operating system Oasis. I can understand why they wouldn't want to go completely with North Star after their experience with P.T.

Well, they just couldn't do it all by themselves! They came up with a fantastic business system using software they had written themselves. But, what small business can afford the

Continued on back page.

UNDERSTANDING AND USING YOUR SOL: OUTPUT ROUTINES

Dear PROTEUS People,

Enclosed you will find about 20 pages of stuff that I have written up for publication in our beloved newsletter. My intent in all this is not to hog space in PROTEUS. I know that we have a very diverse readership. Some readers will find this stuff either useless (because they can do it all by themselves), or unintelligible (because they couldn't care less about such mundane drivel as I/O routines). But, if my hunch is right, there is a significant number of readers who continue to struggle along every time they have to interface something to their SOL.

The first nine pages are in the same format as my previous submission on SOL INPUT ROUTINES, except that this time I am dealing with SOL OUTPUT ROUTINES. The next group of pages comprises an attempt to present a series of PRINTER DRIVER ROUTINES for serial printers. Included are simple and not so simple drivers for TTY, IDS 440 PAPER TIGER, HEATH H14, DIABLO, and printers that use either the CTS or DSR lines for handshaking. Look them over carefully. There are some neat things there, like FORMFEEDS for even klutsy old TTY's, and AUTOPAGING for you folks that like margins at the bottom of a page. If your printer is not reperesented, it is because I don't have any info on it. Send me the necessary information, and I will gladly whip up a custom driver. I would appreciate receiving letters from as many of you as possible, describing how you interfaced your own particular printer, and any problems or neat solutions that you found. Include a listing of your own custom driver if at all possible. I will try to wade through all the stuff I receive, and use the gathered information to enhance the chapters I am writing on SOL I/O. I hate re-inventing the wheel. Please help me in my efforts to continue to provide support for the SOL.

The last two pages are an unabashed advertisement for some software that I have written up for SOL/NORTHSTAR users. There are several versions to support a variety of printer types. The folks who received advance copies of this software were very enthusiastic about it. And the price is right. I would also like to say that the response to my MODIFIER II program has been quite good. The MODIFIER II and SOLO5 programs will continue to be available. The price is $10 for the MODIFIER II program and $15 for SOLO5. SOLO5 comes with SOURCE code. The SOURCE for the MODIFIER II is $10. (Source can be purchased separately only by past customers).

Please specify what kind of system configuration you have when ordering. Specify SOLOS/CUTER tape or 8" soft sector CP/M disk.

Continued on next page.

UNDERSTANDING AND USING YOUR SOL: OUTPUT ROUTINES Cont.

Specify if ordering tape for eventual use on a CP/M disk system... I will then supply it in Richard Greenlaw's TAPEDISK format which make loading a breeze. By the way, NON-SOL systems can run my MODIFIER and SOLO5 programs with only slight modification. I can supply these on tape in Kansas City Standard 3ØØ or 12ØØ Baud, complete with a loader program for reading in the tape. The user only needs to know the address of his tape read-a-byte routine.

Sorry to take up so much valuable space in Proteus, folks. I can only hope that many of you find this stuff useful in one way or another.

Sincerely yours,
Fr. Thomas McGahee
Fr. Thomas McGahee S.D.B.

PS: If you could print this stuff together as a block, it would help. Thanks, Tony.

*** GENERAL OVERVIEW OF OUTPUT ROUTINES ***

SOUT:	**CØ19:**	**Uses pseudoport specified by OPORT (C8Ø7).**	
AOUT:	**CØ1C:**	**Uses pseudoport specified by "A" register.**	
VDMOT:	**CØ54:**	**SOL VIDEO DISPLAY**	**(Pseudoport Ø).**
SDROT:	**CØ4A:**	**SOL SERIAL PORT.**	**(Pseudoport 1).**
PROUT:	**C2E6:**	**SOL PARALLEL PORT.**	**(Pseudoport 2).**
ERROT:	**C2D2:**	**USER-DEFINED OUTPUT "DEVICE".**	**(Pseudoport 3).**
		(Gets address from UOPRT at C8Ø2-C8Ø3).	

CHARACTERISTICS COMMON TO ALL OUTPUT ROUTINES

CHARACTER TO BE SENT IS IN "B": RETURN IS MADE WITH CHARACTER IN "B": ONLY THE "A" REGISTER IS AFFECTED.

```
TYPICAL CALLING PROGRAM:
PROG    MOV B,A         ;TRANSFER CHARACTER TO "B".
        CALL OUTPUT     ;OUTPUT CHARACTER FROM "B".
        *** RETURNS WITH CHARACTER IN "B" ***
```

Some programs, such as CP/M, pass the character to be printed in the "C" register, and require that the character be returned in both "A" and "C", with no other registers affected. One simple program to do just this is now presented.

```
CPMOUT  PUSH B             ;SAVE B,C REGISTER PAIR.
        MOV B,C            ;CP/M HAS IT IN "C", SOL NEEDS IT IN "B".
        CALL OUTPUT        ;OUTPUT CHARACTER FROM "B".
        POP B              ;RESTORE B,C REGISTER PAIR.
        MOV A,C            ;COPY CHARACTER INTO "A" FOR CP/M.
        RET                ;*** CHARACTER IS IN "A" AND "C" ***
```

*** THE MAJOR ROUTINES ***

THE SOUT ROUTINE

SOUT: CØ19: Pseudoport number recovered from OPORT (C8Ø7): Sends out character in "B": Only the "A" register is affected. Default device is video display. RESET and certain errors will cause the default condition.

SOUT is the SYSTEMS OUTPUT routine. This is the prime entry point for normal systems output. It is accessed by a CALL CØ19, and sends out the character in the "B" register. The routine preserves all registers except "A".

SOUT accesses one of four possible output devices, which will be dealt with in detail later. For now, it is enough to say that Ø=the Video Display, 1=the Serial Port, 2=the Parallel Port, and 3=a User-Defined Output Routine. Which of these pseudoports will be used depends upon the contents of a status byte called **OPORT**, located at **C8Ø7**. The binary value from Ø-3 found at this location will be loaded into the "A" register, and then control is passed to the AOUT routine (which is discussed next). The default value at OPORT is Ø, for the SOL video display.

The value at OPORT may be changed in two ways. From the video display, when in the SOL COMMAND MODE, typing SE O=X (where X is a value from Ø-3), will cause X to be the current pseudoport. It should be noted that a RESET will cause the default value of Ø to be loaded again.

The second way to load a value into OPORT is under program control. For instance, the series of instructions MVI A,3 / STA C8Ø7 would cause the User-Defined Output Routine to become current. By the same token, a program can find out what the current pseudoport is by using the instruction LDA C8Ø7.

THE AOUT ROUTINE

AOUT: CØ1C: Enter with pseudoport # in "A": Enter with character to be output in "B": Only the "A" register affected.

AOUT stands for "A-OUTPUT". The AOUT routine is accessed by placing a pseudoport code in the "A" register, the character to be output in the "B" register, and calling CØ1C. The pseudoport code can have the following values:

```
Ø = VDMOT:  CØ54, the SOL VIDEO DISPLAY
1 = SDROT:  CØ4A, the SOL SERIAL PORT
2 = PROUT:  C2E6, the SOL PARALLEL PORT
3 = ERROT:  C2D2, the SOL USER-DEFINED ROUTINE
    (ERROT vectors output to the address found at UOPRT C8Ø2-C8Ø3)
```

When the AOUT routine is called, the value in the "A" register is used to vector to one of the above mentioned pseudoport routines. It is up to the user's program to insure that the "A" register contains a pseudoport number from Ø-3. (The AOUT program will discard all but the two least significant bits, as it contains its own ANI 3 instruction).

*** PSEUDOPORTS ***

BESIDES USING SOUT AND AOUT, PSEUDOPORTS MAY BE DIRECTLY CALLED

Ø: VIDEO DISPLAY: VDMOT: CØ54

Enter with character to be output in "B". Only the "A" register is affected.

The Video Display is pseudoport Ø, the default pseudoport. Besides being available via calls to SOUT and AOUT, it may be directly accessed via a CALL VDMOT (CALL CØ54). It is suggested that either SOUT or AOUT (with "A" set to Ø), be used to access this routine, as these are standard entry points for all SOL systems, but the address

Continued on next page.

of VDMOT may vary. The VDMOT routine processes screen commands as listed below:

```
LEFT-ARROW      81H     CTRL/A  Ø1H     ;MOVE CURSOR LEFT
                                        ; (WRAP MODE).
CLEAR           8BH     CTRL/K  ØBH     ;CLEAR/HOME SCREEN.
RETURN          ØDH     CTRL/M  ØDH     ;ERASE TO END OF LINE
                                        ; GO TO START OF LINE.
HOME CURSOR     8EH     CTRL/N  ØEH     ;HOME CURSOR
RIGHT-ARROW     93H     CTRL/S  13H     ;MOVE CURSOR RIGHT
                                        ; (WRAP MODE).
UP-ARROW        97H     CTRL/W  17H     ;MOVE CURSOR UP.
                                        ; (WRAP MODE).
DOWN-ARROW      9AH     CTRL/Z  1AH     ;MOVE CURSOR DOWN
                                        ; (WRAP MODE).
```

IN ADDITION, THE FOLLOWING ESCAPE SEQUENCES ARE SUPPORTED:

```
ESC CTRL/A XX   ;Place CURSOR at position XX of the current line
                ; where XX is in the range ØØ-3F HEX.

ESC CTRL/B XX   ;Place the cursor at line XX. ØØ is top line.
                ; XX must be in the range ØØ-ØF HEX.

ESC CTRL/C      ;RETURN character position in "B" (ØØ-3FH).
                ; RETURN line position in "C" (ØØ-ØFH).

ESC CTRL/D      ;RETURN memory address of cursor in "BC" pair.

ESC CTRL/E XX or
ESC CTRL/F XX or
ESC CTRL/G XX   ;XX is output to screen EXACTLY AS IS
                ; NO RESTRICTIONS.

ESC CTRL/H XX   ;Display speed set to XX. ØØ=fastest, FF=slowest.

ESC CTRL/I XX   ;SAME AS "ESC CTRL/A XX".
```

VIDEO DISPLAY PORT ASSIGNMENTS AND HOUSEKEEPING

Video Display Status Port=DSTAT=FE
(An XRA A / OUT DSTAT will reset the screen scroll parameters).
Video Display Memory=CCØØ-CFFF organized as 16 lines by 64 characters.
Current Character Position: NCHAR: C8Ø8
Current Line Position: LINE: C8Ø9
Beginning Of Text Displacement: BOT: C8ØA
Display Speed: SPEED: C8ØB

Due to its length and complexity, the VDMOT routine is not shown here. The VDMOT routine does not allow entry of reverse video characters DIRECTLY, but these can be handled using the ESC CTRL/E XX sequence discussed above. A routine that allows reverse video display of all but the special screen control characters, and still allows all normal screen operations including cursor control is given below. It is patterned after VDMOT.

PROGRAM TO ALLOW REVERSE VIDEO (USE AS A CUSTOM ROUTINE)

```
SCRN    PUSH H          ;SAVE REGISTERS.
        PUSH D
        PUSH B
        LDA  ØC8ØCH     ;TEST FOR ESCAPE SEQUENCE.
        ORA A
        JNZ ØC15FH      ;IF ESCAPEFLAG IS ON, PROCESS SPECIAL.
```

```
TBLCHK  MOV A,B         ;NEED IN "A" TO STRIP PARITY BIT.
        STA CHAR        ;***SAVE ORIGINAL FOR LATER.
        ANI 7FH         ;STRIP PARITY BIT.
        MOV B,A         ;SUBROUTINES WANT IT HERE TOO.
        JZ ØCØ6BH       ;IF NULL, DO AN EXIT.
        LXI H,ØC273H    ;POINT TO SPECIAL CHARACTER TABLE.
        CALL SRCH       ;SEARCH TABLE & PROCESS IF FOUND.
        JMP ØCØ6BH      ;***FINISH INSIDE SOLOS.
NXT     INX H           ;SKIP TO NEXT CHAR. ENTRY.
        INX H           ; IGNORING ADDRESSES.

SRCH    MOV A,M         ;GET CHAR. FROM TABLE.
        ORA A
        JZ NOPE         ;IF LAST ENTRY, NOT SPECIAL.
        CMP B           ;TEST FOR A MATCH.
        INX H           ;GET READY FOR NEXT TRY.
        JNZ NXT         ;TRY AGAIN IF NO MATCH.
        PUSH H          ;SAVE ADDRESS IF MATCH FOUND.
        CALL ØC136H     ;REMOVE CURSOR.
        XTHL            ;GET DISPATCH ADDRESS TO HL.
        JMP ØC227H      ;DISPATCH IT.
NOPE    MOV A,B         ;CHECK FOR DELETE.
        CPI 7FH
        RZ              ;DO NOT ECHO DELETE.
        LDA CHAR        ;***RECOVER ORIGINAL CHARACTER.
        MOV B,A         ;SUBROUTINE WANTS IT IN "B".
        JMP ØCØ98H      ;***FINISH WITHIN SOLOS.
CHAR    NOP             ;*RAM STORAGE FOR CHARACTER*
                        ;*MAY BE ANYWHERE IN RAM*
        *** NORMAL OR REVERSE VIDEO CHARACTER NOW PUT ON SCREEN ***
```

1: SERIAL PORT: SDROT: CØ4A

Enter with character to be output in "B". Only the "A" register is affected. ("A"="B" upon return).

The Serial Port is pseudoport 1. Besides being available via calls to SOUT and AOUT, it may be directly accessed via a CALL SDROT (CALL CØ4A). It is suggested that either SOUT or AOUT (with "A" set to 1), be used to access this routine, as these are standard entry points for all SOL systems, but the address of SDROT may vary.

The SOL interface connector for the SERIAL port is connected as if the SOL were itself a terminal, not a computer. This can cause serious problems when connecting to most other terminals, since most people use their SOL as a computer rather than as a terminal. The following pairs of pins on the serial connector should be EXCHANGED:

2 and 3	Transmit and Receive
4 and 5	Request To Send and Clear To Send
6 and 2Ø	Data Set Ready and Data Terminal Ready

The above are EIA RS232C signals. If using a current loop interface, no exchanges are necessary.

It is also interesting to note that many printers can communicate adequately with only 1,2,3, and 7 connected, since not all serial terminals supply or need handshaking signals.

I perform the necessary exchanges by having a short cable with a male on one end, a female on the other, and only the lines 1,2,3,4,5,6,7, and 2Ø connected (with the appropriate switchovers being made at the male connector). This method requires no changes to either the SOL or the terminal, and also provides a longer connector length overall.

Continued on next page.

UNDERSTANDING AND USING YOUR SOL: OUTPUT ROUTINES *Cont.*

J1 PINOUTS FOR SERIAL CONNECTOR (FEMALE DB25S ON SOL)

1	CG	CHASSIS GROUND
2	TD	TRANSMIT DATA (OUT)
3	RD	RECEIVE DATA (IN)
4	RTS	REQUEST TO SEND (OUT)
5	CTS	CLEAR TO SEND (IN)
6	DSR	DATA SET READY (IN)
7	SG	SIGNAL GROUND
8	CD	CARRIER DETECT (IN)
11	CLO	CURRENT LOOP OUT
12	LR1	LOOP RECEIVER 1
13	LR2	LOOP RECEIVER 2
2Ø	DTR	DATA TERMINAL READY (OUT)
23	LCS	LOOP CURRENT SOURCE

The following port information is made available for those who might find it useful.
Serial Status Port=SERST=F8
Status Bit=STBE=bit 7 (HIGH means OK TO SEND)
Serial Data Port=SDATA=F9 (8 bits, normal)
*** it is often wise to strip off the MSB before sending ***

Additional status bits associated with the SERIAL port, but not handled in the SOL software are shown below.

BIT	NAME	FUNCTION	ACTIVE DIRECTION
Ø	SCD	Serial Carrier Detect (EIA)	1=Carrier on
1	SDSR	Serial Data Set Ready (EIA)	Ø=Link OK
2	SPE	Serial Parity Error (UART)	1=Error
3	SFE	Serial Framing Error (UART)	1=Error
4	SOE	Serial Overrun Error (UART)	1=Error
5	SCTS	Serial Clear To Send (EIA)	Ø=Clear
6	SDR	Serial Data Ready (UART)	1=Ready
7	STBE	Serial Xmit Buff Empty (UART)	1=Empty

The SOL SERIAL OUTPUT ROUTINE follows:

```
SDROT   IN SERST        ;GET SERIAL STATUS WORD.
        RAL             ;TEST FOR SERIAL DATA READY.
        JNC SDROT       ;CARRY SET=OK TO SEND.
        MOV A,B         ;WE SEND FROM "A".
        OUT SDATA       ;SEND CHARACTER.
        RET             ;***CHAR. IS IN "A" AND "B".
```

NOTE: instead of ANI STBE, RAL was used since it saved 1 byte.

2: PARALLEL PORT: PROUT: C2E6

Enter with character to be output in "B". Only the "A" register is affected. ("A"="B" upon return).

The Parallel Port is pseudoport 2. Besides being available via calls to SOUT and AOUT, it may be directly accessed via a CALL PROUT (CALL C2E6). It is suggested that either SOUT or AOUT (with "A" set to 1), be used to access this routine, as these are standard entry points for all SOL systems, but the address of PROUT may vary.

PARALLEL PORT ASSIGNMENTS

Parallel Port Status=STAPT=FA
Status Bit=PXDR=bit 2 (LOW means OK TO SEND)
Parallel Data Port=PDATA=FD

NOTE: Status Port FA is also used by the Keyboard and Cassette Interface.

J2 PINOUTS FOR PARALLEL CONNECTOR (MALE DB25P ON SOL)

1	CG	CHASSIS GROUND		
2	SG	SIGNAL GROUND		
3	IE	OUTPUT ENABLE	(IN)	1=ENABLED
4	*DR	*DATA READY	(IN)	Ø=READY
5	*IAK	*OUTPUT ACKNOWLEDGE	(OUT)	Ø=ACKNOWLEDGE
6	ID7	OUTPUT DATA, BIT 7	(IN)	
7	ID6	OUTPUT DATA, BIT 6	(IN)	
8	ID5	OUTPUT DATA, BIT 5	(IN)	
9	ID4	OUTPUT DATA, BIT 4	(IN)	
1Ø	ID3	OUTPUT DATA, BIT 3	(IN)	
11	ID2	OUTPUT DATA, BIT 2	(IN)	
12	ID1	OUTPUT DATA, BIT 1	(IN)	
13	IDØ	OUTPUT DATA, BIT Ø	(IN)	
14	US	UNIT SELECT	(OUT)	
15	OE	OUTPUT ENABLE	(IN)	1=ENABLED
16	*XDR	*EXTERNAL DEVICE READY	(IN)	Ø=READY
17	*OL	*OUTPUT LOAD	(OUT)	Ø=LOAD
18	OD7	OUTPUT DATA, BIT 7	(OUT)	
19	OD6	OUTPUT DATA, BIT 6	(OUT)	
2Ø	OD5	OUTPUT DATA, BIT 5	(OUT)	
21	OD4	OUTPUT DATA, BIT 4	(OUT)	
22	OD3	OUTPUT DATA, BIT 3	(OUT)	
23	OD2	OUTPUT DATA, BIT 2	(OUT)	
24	OD1	OUTPUT DATA, BIT 1	(OUT)	
25	ODØ	OUTPUT DATA, BIT Ø	(OUT)	

Many of the OUTPUT lines have active pullup. For the simplest case Parallel Output with HANDSHAKING, the following hardware arrangement can be used.

1,2 BOTH connected to Parallel Output Device GROUND.
14,15 Leave floating HIGH.
16 Parallel Output Device must SUPPLY ACTIVE LOW signal. (Should go LOW when SOL MAY SEND DATA).
17 Parallel Output Device must sample this line. (Whenever it is LOW, DATA IS AVAILABLE ON DATA LINES).
18-25 Parallel Output Device should RECEIVE DATA on these lines.

The following is the SOL PARALLEL OUTPUT ROUTINE

```
PROUT   IN STAPT        ;GET STATUS WORD.
        ANI PXDR        ;TEST PARALLEL PORT ONLY.
        JNZ PROUT       ;WAIT UNTIL OK TO SEND.
        MOV A,B         ;SEND FROM "A".
        IUT PDATA       ;SEND CHARACTER.
        RET             ;*** CHAR. IS IN "A" AND "B" ***
```

3 USER-DEFINED PORT: ERROT: C2D2

Enter with character to be output in "B". Only the "A" register is affected. Any OTHER characteristics are up to the USER, but THESE CHARACTERISTICS MUST BE MET!

The CUSTOM or USER-DEFINED PORT is pseudoport 3. Besides being available via calls to SOUT and AOUT, it may be directly accessed via a CALL ERROT (CALL C2D2), or by means of a direct call to the address of the user routine. It is suggested that either SOUT or AOUT (with "A" set to 3), be used to access this routine, as these are standard entry points for all SOL systems, but the address of ERROT may vary.

Continued on next page.

NOTE: Use of pseudoport 3 assumes that the user has placed an output routine in memory that meets the above requirements. In addition, the address of the CUSTOM routine must be loaded into memory at **UOPRT (C8Ø2-C8Ø3)**. There are two ways of accomplishing this: The user may type SE CO=XXXX, where XXXX is the address of the user-defined CUSTOM routine, or he may load the address in under program control. For example, the instruction sequence LXI H,1234 / SHLD C8Ø2 would cause the address 1234 to be loaded at C8Ø2-C8Ø3. (Addresses are loaded in reverse-order... Low order'at C8Ø2, and High order at C8Ø3).

Remember also that it is up to the user to make the current pseudoport=3 when using SOUT, and to make "A"=3 when using AOUT to access the CUSTOM Output Routine. After a RESET, the OPORT and UOPRT are set to ØØ and ØØØØ repectively. The ERROT routine will assume an error has occurred whenever it finds ØØØØ at UOPRT. This means that ØØØØ may never be used as the address of a CUSTOM routine.

The CUSTOM output routine allows the user to write output routines to handle special devices, and even include special character checks and conversions within the CUSTOM routine. The user can often put this ability to good use. I have, for instance, written CUSTOM output drivers that automatically keep track of how many linefeeds have occurred, and when the proper number is detected, printing is halted until the space bar is hit. This allows me to use sheet paper. CUSTOM routines are a good way to intercept certain characters. For instance, a FORMFEED character might be used to halt printing. This would allow a SELECTRIC with sheet paper to run with software written for another style printer with roll paper.

Also note that it is often useful to have certain CUSTOM routines automatically switch the current pseudoport to Ø when a given process has been completed. The CUSTOM routines are often much more than just simple output drivers! Their use is limited only by the imagination and skill of the user.

More than one custom routine may reside in memory at once, but only one custom routine at a time may be called in the flexible manner allowed by SOUT and AOUT. This is not really a problem, since control characters may be used to allow one CUSTOM routine to make another custom routine current. I routinely use CONTROL/X as a special character that causes any given CUSTOM routine to load in another custom routine. You have to design your own CUSTOM routines to meet your own particular needs, but believe me, the ability to access such programs using SOUT and AOUT is a powerful feature.

This material presented courtesy of:

Fr. Thomas McGahee
2Ø2 Union Ave.
Paterson, NJ Ø75Ø2
(2Ø1-595-88ØØ)

```
*** SAMPLE DRIVERS FOR IDS 44Ø PAPER TIGER ***
*** ASSUMES YOU HAVE IT CONNECTED TO SERIAL PORT ***
**
*** WRITTEN BY FR. THOMAS MCGAHEE
*** DON BOSCO TECHNICAL HIGH SCHOOL
*** 2Ø2 UNION AVE., PATERSON, NJ Ø75Ø2
***      JUNE 9,198Ø

********** EQUATE TABLE **********

PAGE     EQU     5Ø       ;FIFTY LINES PRINTED PER PAGE.
MAX      EQU     66       ;MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).
FRMFD    EQU     ØCH      ;CODE FOR FORMFEED.
SERST    EQU     ØF8H     ;SERIAL STATUS PORT.
SDROT    EQU     ØCØ4AH   ;SERIAL OUTPUT ROUTINE.
VDMOT    EQU     ØCØ54H   ;VIDEO OUTPUT ROUTINE.
LF       EQU     ØAH      ;CODE FOR LINE-FEED.
CR       EQU     ØDH      ;CODE FOR CARRIAGE-RETURN.

*** FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER ***
*** INCLUDES HANDSHAKING. ENTER WITH CHARACTER IN "B" ***
*** EXIT WITH CHARACTER IN "A" AND "B" ***

TIGER:   IN SERST         ;CHECK IF PRINTER READY.
         ANI 22H          ;CHECK BOTH CTS AND DSR.
         JNZ TIGER        ;WAIT IF BUSY.
         CALL SDROT       ;PRINT CHARACTER IN "B".
         RET              ;RETURN.

*** SECOND EXAMPLE SHOWS A LIST DRIVER FOR CP/M. ***
*** THIS ONE ENTERS WITH CHARACTER IN "C" ***
*** EXIT IS WITH CHARACTER IN BOTH "A" AND "C". ***
*** THIS ALSO INCLUDES AUTOMATIC PAGING. ***
*** AFTER EVERY 5Ø LINEFEEDS A FORMFEED OCCURS. ***
*** THIS KEEPS LISTINGS NEAT. ***
*** ANOTHER ADDED FEATURE IS THAT THE VIDEO SCREEN ***
*** WILL ECHO WHAT IS BEING PRINTED. ***

LIST:    PUSH B           ;SAVE B,C.
         MOV B,C          ;CP/M HAS IT IN "C" & SOL NEEDS IT IN "B".
VIDEO:   CALL VDMOT       ;PRINT TO VIDEO FIRST.
*** ELIMINATE "CALL VDMOT" IF YOU WANT PRINTER ONLY.
*** I LIKE TO SEE THE STUFF ON THE SCREEN AS WELL.
PRNTR:   IN SERST         ;CHECK IF PRINTER READY.
         ANI 22H          ;CHECK BOTH CTS AND DSR.
         JNZ PRNTR        ;WAIT UNTIL CLEAR.
         MOV A,B
         CPI FRMFD        ;CHECK FOR FORMFEED.

         JZ FFEED
PRINT:   CALL SDROT       ;PRINT USING SERIAL PORT.
         CPI LF           ;WAS IT A LINE-FEED?
         RNZ
PAGER:   LDA COUNT        ;GET CURRENT COUNT.
         DCR A            ;REDUCE IT BY 1.
         STA COUNT        ;STORE NEW COUNT.
         JNZ NOPAGE       ; IGNORE IF NOT END.
FFEED:   LDA LENGTH       ;SET UP NEW PAGE LENGTH.
         STA COUNT
         MVI B,FRMFD      ;PRINT FORMFEED.
         CALL SDROT       ;DO A FORMFEED.
NOPAGE:  POP B            ;RECOVER ORIGINAL B,C.
         MOV A,C          ;CP/M LIKES IT IN "A" TOO.
         RET

COUNT    DB       PAGE     ;LINES LEFT ON THIS PAGE.
LENGTH   DB       PAGE     ;# OF PRINTED LINES PER PAGE
*** NOTE THAT "PAGE" MUST BE < THAN OR = TO "MAX".

A>

*** SAMPLE DRIVERS FOR DIABLO PRINTERS ***
*** ASSUMES YOU HAVE IT CONNECTED TO THE SERIAL PORT. ***
**
*** WRITTEN BY FR. THOMAS MCGAHEE
*** DON BOSCO TECHNICAL HIGH SCHOOL
*** 2Ø2 UNION AVE., PATERSON, NJ. Ø75Ø2

********** EQUATE TABLE **********

PAGE     EQU     5Ø       ;FIFTY LINES PRINTED PER PAGE.
MAX      EQU     66       ;MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).
```

Continued on next page.

UNDERSTANDING AND USING YOUR SOL: OUTPUT ROUTINES Cont.

```
FRMFD   EQU     ØCH     ;CODE FOR FORMFEED.
SERST   EQU     ØF8H    ;SERIAL STATUS PORT.
SDROT   EQU     ØCØ4AH  ;SERIAL OUTPUT ROUTINE.
VDMOT   EQU     ØCØ54H  ;VIDEO OUTPUT ROUTINE.
SSTAT   EQU     ØCØ42H  ;SERIAL INPUT ROUTINE.
LF      EQU     ØAH     ;CODE FOR LINE-FEED.
CR      EQU     ØDH     ;CODE FOR CARRIAGE-RETURN.
ACK     EQU     Ø6H     ;ACKNOWLEDGE CODE.
ETX     EQU     Ø3H     ;END-OF-TEXT CODE.
BUFF    EQU     8Ø      ;8Ø CHARACTERS= 1/2 OF BUFFER.

*** FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER. ***
*** INCLUDES HANDSHAKING VIA ETX AND ACK CODES. ***
*** ENTER WITH CHARACTER IN "B". ***
*** EXIT WITH CHARACTER IN "A" & "B". ***

DIABLO: LDA CHRCNT      ;UPDATE CHARACTER COUNT.
        DCR A
        STA CHRCNT      ;STORE UPDATED CHARACTER COUNT.
        JNZ OK          ;IF OK, ACTIVATE PRINTER.
        MVI A,BUFF      ;OTHERWISE RESET CHARACTER COUNT.
        STA CHRCNT
        PUSH B          ;SAVE CHARACTER.
        MVI B,ETX       ;PREPARE TO SEND ETX.
        CALL SDROT      ;SEND IT.
WAIT:   CALL SSTAT      ;CHECK FOR ACK.
        JZ WAIT         ;WAIT UNTIL A CHARACTER IS READY.
        ANI 7FH         ;STRIP OFF MSB.
        CPI ACK         ;CHECK TO SEE IF IT IS ACK.
        JNZ WAIT        ;IGNORE ANYTHING ELSE.
        POP B           ;RECOVER ORIGINAL CHARACTER.
OK:     CALL SDROT      ;PRINT ORIGINAL CHARACTER.
        RET             ;DONE.

CHRCNT  DB      BUFF    ;CHARACTER COUNT.

*** THE NEXT ROUTINE IS A CP/M LIST DRIVER. ***
*** ENTER WITH CHARACTER IN "C". ***

*** EXIT WITH CHARACTER IN BOTH "A" AND "C". ***
*** INCLUDES VIDEO ECHO AND AUTOPAGING. ***

LIST:   PUSH B          ;SAVE ORIGINAL.
        MOV B,C         ;CP/M HAS IT IN "C", SOLOS NEEDS IT IN "B".
        CALL VDMOT      ;***REMOVE IF VIDEO ECHO NOT WANTED.
        MOV A,B         ;CHECK FOR FORMFEED.
        CPI FRMFD
        JZ FFEED        ;HANDLE FORMFEEDS SPECIAL.
        LDA CHRCNT      ;UPDATE CHARACTER COUNT.
        DCR A
        STA CHRCNT      ;STORE UPDATED CHARACTER COUNT.
        JNZ OK          ;OK IF BUFFER NOT EXCEEDED.
        MVI A,BUFF      ;RESET CHARACTER COUNT
        STA CHRCNT      ; TO BEGINNING.
        MVI B,ETX       ;SEND OUT ETX.
        CALL SDROT
WAIT:   CALL SSTAT      ;WAIT UNTIL ACK RECEIVED.
        JZ WAIT
        ANI 7FH
        CPI ACK
        JNZ WAIT        ;IGNORE ALL BUT ACK.
        MOV B,C         ;GET ORIGINAL BACK INTO "B".
OK:     CALL SDROT      ;PRINT ORIGINAL CHARACTER.
        CPI LF          ;TIME TO END THIS PAGE?
        POP B           ;GET READY FOR A RETURN.
        RNZ             ;RETURN IF NOT A LINE-FEED.
PAGER:  LDA COUNT       ;UPDATE/CHECK COUNT.
        DCR A
```

```
        STA COUNT       ;SAVE UPDATED COUNT.
        MOV A,C         ;GET READY FOR RETURN.
        RNZ             ;RETURN IF NOT END OF PAGE.
        PUSH B          ;KEEP A CLEAN STACK!
FFEED:  LDA PAGE        ;RESET PAGE COUNT.
        STA COUNT
        MVI B,FRMFD     ;UNCONDITIONAL FORMFEED SENT.
        CALL SDROT
        POP B           ;RESTORE ORIGINALS.
        MOV A,C         ;CP/M WANTS IT IN "A" TOO.
        RET


CHRCNT  DB      BUFF    ;CHARACTER COUNT FOR DIABLO.
COUNT   DB      PAGE    ;LINES PER PAGE (PRINTED).
LENGTH  DB      PAGE    ;MAXIMUM NUMBER ALLOWED TO PRINT.


*** SAMPLE DRIVERS FOR HEATH H14 PRINTERS ***
*** ASSUMES YOU HAVE IT CONNECTED TO THE SERIAL PORT. ***
**
*** WRITTEN BY FR. THOMAS MCGAHEE
*** DON BOSCO TECHNICAL HIGH SCHOOL
*** 2Ø2 UNION AVE., PATERSON, NJ. Ø75Ø2

********** EQUATE TABLE **********

PAGE    EQU     5Ø      ;FIFTY LINES PRINTED PER PAGE.
MAX     EQU     66      ;MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).
FRMFD   EQU     ØCH     ;CODE FOR FORMFEED.
SERST   EQU     ØF8H    ;SERIAL STATUS PORT.
SDROT   EQU     ØCØ4AH  ;SERIAL OUTPUT ROUTINE.
VDMOT   EQU     ØCØ54H  ;VIDEO OUTPUT ROUTINE.
SSTAT   EQU     ØCØ42H  ;SERIAL INPUT ROUTINE.
LF      EQU     ØAH     ;CODE FOR LINE-FEED.
CR      EQU     ØDH     ;CODE FOR CARRIAGE-RETURN.
BUSY    EQU     13H     ;PRINTER BUSY CODE.
READY   EQU     11H     ;PRINTER READY CODE.

*** FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER. ***
*** INCLUDES HANDSHAKING VIA BUSY AND READY CODES. ***
*** ENTER WITH CHARACTER IN "B". ***
*** EXIT WITH CHARACTER IN "A" & "B". ***

H14:    CALL SSTAT      ;CHECK TO SEE IF BUSY FIRST.
        JZ PRINT        ;IF NOT, THEN PRINT IT.
BSY:    CPI BUSY        ;IS IT REALLY BUSY?
        JNZ PRINT       ;IF FALSE ALARM, PRINT "B".
RDY:    CALL SSTAT      ;WAIT FOR READY SIGNAL.
        JZ RDY
        CPI READY       ;CHECK TO SEE IF IT IS READY.
        JNZ RDY         ;IGNORE ANYTHING ELSE.
PRINT:  CALL SDROT      ;PRINT ORIGINAL CHARACTER.
        RET             ;DONE.

*** THE NEXT ROUTINE IS A CP/M LIST DRIVER. ***
*** ENTER WITH CHARACTER IN "C". ***
*** EXIT WITH CHARACTER IN BOTH "A" AND "C". ***
*** INCLUDES VIDEO ECHO AND AUTOPAGING. ***

LIST:   PUSH B          ;SAVE ORIGINALS.
        MOV B,C         ;CP/M HAS IT IN "C", SOLOS NEEDS IT IN "B".
        CALL SSTAT      ;CHECK TO SEE IF BUSY FIRST.
        JZ PRINT        ;IF NOT, THEN PRINT IT.
BSY:    CPI BUSY        ;IS IT REALLY BUSY?
        JNZ PRINT       ;IF FALSE ALARM, PRINT "B".
RDY:    CALL SSTAT      ;WAIT FOR READY SIGNAL.
```

Continued on next page.

```
          JZ RDY
          CPI READY       ;CHECK TO SEE IF IT IS READY.
          JNZ RDY         ;IGNORE ANYTHING ELSE.
PRINT:    MOV A,B         ;PERFORM FORMFEED CHECK.
          CPI FRMFD
          JZ FFEED        ;FORMFEEDS ARE HANDLED SPECIAL.
          CALL SDROT      ;PRINT ALL OTHERS.
          CPI LF
          POP B           ;PREPARE FOR RETURN.
          RNZ             ;RETURN IF NOT A LINE-FEED.
          LDA COUNT       ;UPDATE LINE COUNT.
          DCR A
          STA COUNT       ;STORE UPDATED COUNT.
          MOV A,C         ;CP/M LIKES IT IN "A" TOO.
          RNZ             ;RETURN IF PAGE NOT DONE.
          PUSH B          ;KEEP STACK CLEAN!
FFEED:    LDA LENGTH      ;RESET PAGE COUNT.
          STA COUNT
          MVI B,FRMFD     ;SEND UNCONDITIONAL FORMFEED.
          CALL SDROT
          POP B           ;RECOVER ORIGINALS.
          RET             ;ALL DONE.

COUNT     DB      PAGE    ;LINES PER PAGE (PRINTED).
LENGTH    DB      PAGE    ;MAXIMUM NUMBER ALLOWED TO PRINT.
```

```
*** SAMPLE DRIVERS FOR TTY-LIKE PRINTERS ***
*** PROVIDES FOR NULLS AFTER EACH LINE FEED. ***
*** ASSUMES YOU HAVE IT CONNECTED TO THE SERIAL PORT. ***
**
*** WRITTEN BY FR. THOMAS MCGAHEE
*** DON BOSCO TECHNICAL HIGH SCHOOL
*** 2Ø2 UNION AVE., PATERSON, NJ. Ø75Ø2

********** EQUATE TABLE **********

PAGE    EQU     5Ø      ;FIFTY LINES PRINTED PER PAGE.
MAX     EQU     66      ;MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).
FRMFD   EQU     ØCH     ;CODE FOR FORMFEED.
SDROT   EQU     ØCØ4AH  ;SERIAL OUTPUT ROUTINE.
VDMOT   EQU     ØCØ54H  ;VIDEO OUTPUT ROUTINE.
LF      EQU     ØAH     ;CODE FOR LINE-FEED.
NULLS   EQU     1Ø      ;NUMBER OF NULLS AFTER LF.

*** FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER. ***
*** ENTER WITH CHARACTER IN "B". ***
*** EXIT WITH CHARACTER IN "A" & "B". ***
*** SENDS OUT NULLS FOLLOWING A LINE-FEED. ***
*** FORMFEED ACCOMPLISHED VIA CALCULATED LINE-FEEDS. ***


TTY:    MOV A,B         ;CHECK FOR FORMFEED.
        CPI FRMFD
        JZ FFEED        ;HANDLE FORMFEEDS SPECIAL.
        CALL SDROT      ;PRINT CHARACTER IN "B".
        CPI LF          ;THEN CHECK TO SEE IF IT WAS A LINE-FEED.
        RNZ             ;IF NOT, WE IS ALL DONE.
        CALL NULL       ;IF SO, SEND OUT SOME NULLS!
PAGER:  LDA COUNT       ;UPDATE LINE COUNT.
        DCR A
        STA COUNT       ;STORE UPDATED LINE COUNT.
        MOV A,B         ;PREPARE FOR RETURN.
        RNZ             ;RETURN IF ALL DONE.
```

```
        MVI A,MAX       ;RESET LINE COUNT AT PAGE BOUNDARY.
        STA COUNT
        MOV A,B
        RET             ;RETURN WITH NEW PAGE SET.
FFEED:  LDA COUNT       ;GET CURRENT PAGE LENGTH.
        PUSH B          ;SAVE ORIGINALS.
LFLUP:  PUSH PSW        ;SAVE # OF LINES TO DO.
        MVI B,LF        ;SEND OUT A LINEFEED.
        CALL SDROT
        CALL NULL       ;APPEND THE NECESSARY NULLS TOO.
        POP PSW         ;RECOVER # OF LINES TO GO.
        DCR A           ;UPDATE IT.
        JNZ LFLUP       ;DO MORE IF NEEDED.
        POP B           ;RECOVER ORIGINALS.
        MVI A,MAX       ;RESET FOR A FULL PAGE LENGTH.
        STA COUNT
        MOV A,B         ;COPY "B" INTO "A" FOR CONSISTENCY.
        RET             ;AND NOW WE IS DONE.
*** NULL GENERATOR ***
NULL:   PUSH B          ;SAVE ORIGINALS.
        MVI B,Ø         ;"B" CONTAINS THE NULL CHARACTER NOW.
        MVI C,NULLS     ;"C" CONTAINS THE NULL COUNT.
NLOOP:  CALL SDROT      ;SEND OUT A NULL.
        DCR C           ;UPDATE NULL COUNT.
        JNZ NLOOP       ;SEND OUT PROPER AMOUNT.
        POP B           ;RESTORE ORIGINALS.
        RET             ;ON WITH THE SHOW.

COUNT   DB      MAX     ;ALLOWS CALCULATED FORMFEEDS.

*** THE NEXT ROUTINE IS A CP/M LIST DRIVER. ***
*** ENTER WITH CHARACTER IN "C". ***
*** EXIT WITH CHARACTER IN BOTH "A" AND "C". ***
*** INCLUDES VIDEO ECHO AND AUTOPAGING. ***

LIST:   PUSH B          ;SAVE ORIGINALS.
        MOV B,C         ;CP/M HAS IT IN "C", SOLOS NEEDS IT IN "B".
        CALL VDMOT      ;*** REMOVE IF NO VIDEO ECHO DESIRED.
        MOV A,B         ;CHECK FOR FORMFEED.
        CPI FRMFD
        JZ FFEED        ;HANDLE FORMFEEDS SPECIAL.
        CALL SDROT      ;PRINT CHARACTER IN "B".
        CPI LF          ;THEN CHECK TO SEE IF IT WAS A LINE-FEED.
        POP B           ;PREPARE FOR RETURN.
        RNZ             ;IF NOT, WE IS ALL DONE.
        CALL NULL       ;IF SO, SEND OUT SOME NULLS!
PAGER:  LDA COUNT       ;UPDATE LINE COUNT.
        DCR A
        STA COUNT       ;STORE UPDATED LINE COUNT.
        POP B
        MOV A,B         ;PREPARE FOR RETURN.
        RNZ             ;RETURN IF ALL DONE.
FFEED:  LDA COUNT       ;GET CURRENT PAGE LENGTH.
        ADI MAX-PAGE    ;CALCULATE # OF LINES NEEDED.
LFLUP:  PUSH PSW        ;SAVE # OF LINES TO DO.
        MVI B,LF        ;SEND OUT A LINEFEED.
        CALL SDROT
        CALL NULL       ;APPEND THE NECESSARY NULLS TOO.
        POP PSW         ;RECOVER # OF LINES TO GO.
        DCR A           ;UPDATE IT.
        JNZ LFLUP       ;DO MORE IF NEEDED.
        POP B           ;RECOVER ORIGINALS.
```

Continued on next page.

UNDERSTANDING AND USING YOUR SOL: OUTPUT ROUTINES

```
        MVI A,PAGE      ;RESET FOR PROPER PRINT LENGTH.
        STA COUNT
        MOV A,B         ;COPY "B" INTO "A" FOR CONSISTENCY.
        RET             ;AND NOW WE IS DONE.
*** NULL GENERATOR ***
NULL:   PUSH B          ;SAVE ORIGINALS.
        MVI B,Ø         ;"B" CONTAINS THE NULL CHARACTER NOW.
        MVI C,NULLS     ;"C" CONTAINS THE NULL COUNT.
NLOOP:  CALL SDROT      ;SEND OUT A NULL.
        DCR C           ;UPDATE NULL COUNT.
        JNZ NLOOP       ;SEND OUT PROPER AMOUNT.
        POP B           ;RESTORE ORIGINALS.
        RET             ;ON WITH THE SHOW.

COUNT   DB      PAGE    ;ALLOWS CALCULATED FORMFEEDS.
                        ; AND ALSO AUTOPAGING.
```

ADVERTISEMENTS

ADVERTISEMENT: I have custom USER routines for NORTHSTAR that include drivers for a wide variety of printers operating under control of a SOL-2Ø. Each includes the following features:

FULL INSTRUCTIONS AND ASSEMBLY LISTING INCLUDED.
MODE SELECT is converted to Control/C.
LEFT-ARROW is converted to BACKSPACE
RIGHT-ARROW is converted to BACKSPACE
BACKSPACE works on BOTH VIDEO SCREEN and PRINTER.
LOAD is converted to a COMMA for numeric keypad use.
USER MAY EASILY ADD HIS OWN CONVERSIONS.
CONTROL/P can be used to cause PRINTER to ECHO VIDEO!
CONTROL/V can be used to return to VIDEO-ONLY mode.
CONTROL/V/P ARE AVAILABLE AT ALL TIMES, EVEN DURING LISTINGS!
OUTPUT DEVICE may be selected as device #Ø-3.
DEVICE #3 is defined as BOTH VIDEO and PRINTER.
PAGING on all PRINTER output is automatic.
PAGING can be patched to support single-sheet paper.
FORMFEED works even with TTY-type terminals!
PRINTING MAY BE PAUSED.
PRINTER can be switched on and off even during a listing.

PRINTERS SUPPORTED

SERIAL TELETYPE WITH NULLS AFTER LINEFEED
NSTARNUL Connects via SOL Serial Port #1. Provides delay following a linefeed by sending out a user-defined number of nulls. Comes set for 1Ø nulls. FORMFEED is accomplished with linefeeds. MAY BE USED WITH ANY SERIAL PRINTER.

SERIAL WITH HANDSHAKING VIA CLEAR-TO-SEND
NSTARCTS Uses SOL Serial Port #1, and samples status via pins 5 & 6 of J1. This allows operation with any printer that communicates ready status using either pin 4 or 2Ø of the RS232 connector. Use for PAPER TIGER, etc.

SLOWED SERIAL via SOL Serial Port #1.
NSTARSER has no handshaking, but it does allow the user to specify a small delay following EVERY character, and longer delays after LF. Use this version for serial printers that tend to overflow their buffers. The small delay can be "tweaked" by the user to allow serial printers to run at high baud rates without loss of data. FORMFEED is accomplished with linefeeds.

SERIAL HEATH H14 via SOL Serial Port #1.
NSTARH14 INCLUDES HANDSHAKING using Control/Q and Control/S. Allows operation at ANY baud rate without loss of characters. Use with any printer that uses this form of handshaking.

SERIAL DIABLO 162Ø (or equivalents such as IPSI 1622)
NSTARDBL Connects via SOL Serial Port #1. Allows use at ANY baud rate without loss of data. Uses ETX (Ø3) and ACK (Ø6) control codes for handshaking, so will work with ANY serial terminal that uses ETX and ACK codes. Makes full use of buffer.

OTHER TERMINALS including PARALLEL types:
Send complete information on your printer needs including a stamped self-addressed envelope, and I will quote a price for customizing a USER routine specifically for your terminal. New routines are constantly being added, so chances are good that I will already have one to meet your needs. If you have a copy of a simple printer routine that works with your printer, I can use that as a guide.

I also provide other customization services. If you have a specific need, write and let me know. Please include a stamped self-addressed envelope with all enquiries. If you belong to a computer club, or know of other SOL users who might benefit from any of my software, I would appreciate it if you would let them know about what we have to offer. I operate on a shoestring budget here, and rely upon announcements in club newsletters, and word of mouth to provide advertising.

Currently stocked USER routines sell for $1Ø in continental USA, $15 elsewhere, including CANADA. Foreign orders usually shipped airmail, all other are sent First Class. Routines written for printers not currently stocked cost $15 in continental USA, $2Ø elsewhere.

As currently written, DOS is assumed to have an origin at 2ØØØ with the USER AREA starting at 29ØØ. If your system is different, CLEARLY state where the USER AREA should start, and include the locations where the JUMP table entries for COUT, CIN, TINIT, and CONTC are located (they start at 2ØØD in a standard system). CUSTOM ORIGINS require me to reassemble the program and run off a custom manual. Please add $5 to cover my additional costs. (This $5 fee is ALREADY included for those who request a custom printer routine).

Put in plain English, the program costs $1Ø. Add $5 if outside continental U.S.A., and $5 if it includes customizing of any kind.

Program is supplied on SOLOS/CUTER 12ØØ baud cassette tape in BINARY FORMAT. The SOURCE CODE is provided as HARD COPY.

Send check or money order to:
Fr Thomas McGahee
2Ø2 Union Ave.
Paterson, New Jersey Ø75Ø2

For inquiries of any kind concerning this or any of my other software or services, please include a stamped, self-addressed envelope.

REPLACE MONITOR 27Ø8 WITH 2716

Gentlemen:

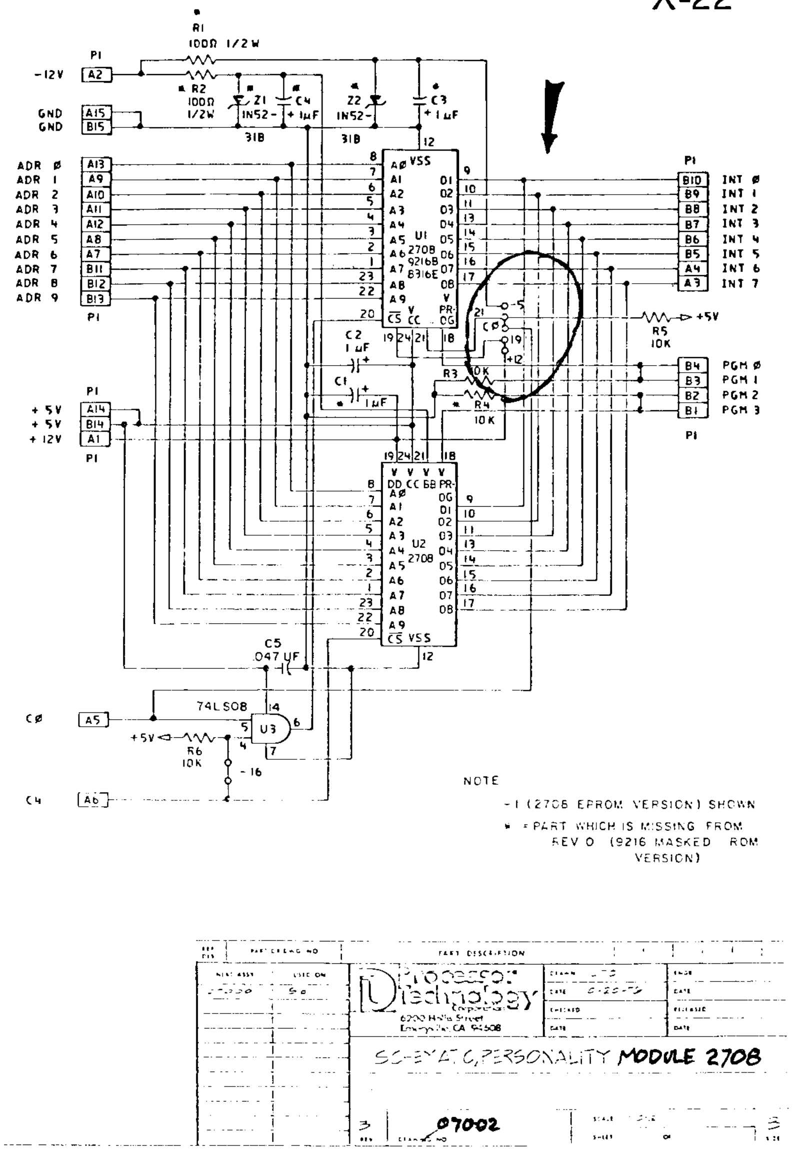
The Sol 20 Monitor Rom (Solos-1) may be easily replaced with a 2716 Eprom. Enclosed are the hardware modifications on the PC board necessary to do this. The Object Code located at C000-C7FF is burned into the Eprom prior to making the hardware changes, or can be done on a second piece of equipment.

Sincerely,

Howard Chace

Howard E. Chace

X-22



REPLACE MONITER ROM WITH 2716 EPROM

1. REMOVE TRACE CONNECTING PIN 19 TO 12V
2. REMOVE TRACE CONNECTING PIN 21 TO CØ
3. INSTALL JUMPER CONNECTING PIN 19 TO CØ
4, INSTALL JUMPER CONNECTING PIN 21 TO 5VDC

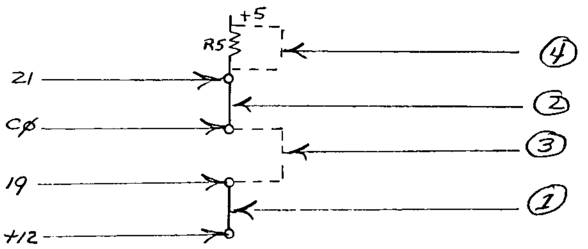


IMAGE FILES/CASSETTE TO HELIOS FIX/MODEM MOD./WORD WIZARD

Dear Stan:

I got my Proteus/News today and that prompted me to write. Other members have expressed this too: I am enjoying growing into my Sol/Helios. I've had it about a year and a half and am just beginning to use some things like the debugger. The more I learn about it, the more I appreciate the system. I have resisted switching to CP/M because, like other writers in the last newsletter, I think PTDOS is so much better. I think the CP/M emulator you mentioned would be great. We could continue with PTDOS but still have access to all the CP/M software available.

I feel I am finally getting to the point where I can provide other Sol users with some answers and not just questions. In my last letter, I raised the question about how to transfer PTC cassette software to Helios disk. When I tried it I was getting bad image files. Now, with some help from Computer Port, I can provide the answer.

I don't completely understand the cause of the problem. It has something to do with the software being designed to work with CUTTER as well as SOLOS and when it's transfered to disk, the wrong value is in register HL.

I do know that the fix works. It is as follows:

In SOLOS type:

```
>GET filename (filename t addr count)
```

SOLOS responds as shown in parentheses above. Then while still in SOLOS type:

```
>EN count <CR>
:21 00 C0 C3 00 00/<CR>
```

Then go to PTDOS and type:

```
*IMAGE filename start addr,count+6,count
```

Example:

```
>GET ZING ZING G 0000 046B
>EN 046B <CR>
:21 00 C0 C3 00 00/ <CR>

*IMAGE ZING,0,0471,046B <CR>
```

The resulting image file will begin execution at the start of the patch and then jump to the start of the original program after the fix is made.

A few months ago I got a modem. Having communications capability opens up whole new areas to explore. The modem I have is the PENRIL that operates at both 300 and 1200 baud. Operating at 300 baud seems so slow when I am accustomed to the SOL flashing things on the screen. However, I sometimes found myself staying at low speed because of the inconvenience (also felt by Frank Sanders) of lifting the monitor, removing the cover and reaching inside the SOL to change the baud rate DIP switches.

Finally, I worked out a way to put the baud rates under software control. After thinking about this problem for some time, I realized that I had a second cassette motor relay sitting in there not doing anything. Also, with HELIOS, I probably never would want hook up two tape recorders. So, I thought, why not hook the relay into the baud rate switches and use that to change from 300 baud to 1200 baud.

I tore my SOL apart, soldered on three wires and a different relay, put it all back together and it worked. I thought it was fantastic to be able to control everything from the keyboard. I can not only switch the SOL speed from the keyboard, I can even switch the modem speed too.

The existing Relay K2 is SPST (single pole, single throw). It must be replaced with a SPDT (single pole, double throw) relay so that one position can be 300 baud and the other can be 1200. The replacement part is SIGMA 191TE1C1-5S. The original relay has a 500 ohm coil resistance. The new one has a 200 ohm coil resistance. This means that IC U97 feeding the relay has to work a little harder. So far this doesn't seem to be a problem.

After you get the old relay out, you must cut the trace connecting pin 1 and 14 next to pin 1. Then install the new relay.

Then, with 3 new 24 ga. wires, on the bottom side of the PC board connect relay pin 1 to the off (front) side of baud rate switch 6. Connect relay pin 8 to the off side of baud rate switch 4. Connect relay pin 14 to the on (rear) side of baud rate switch 8. Actually this last wire can go to any of the on side positions of the baud rate switches. The PC board ties them all together, but 8 is nearest to the relay.

Now, the baud rate switches must all be left in the off position. The default condition, when the SOL is first turned on is 1200 baud. When the relay is activated, it changes the SOL to 300 baud. If you want 300 baud to be the default condition, reverse the two wires at baud rate switches 4 & 6.

The only hardware change needed to switch the modem too was to install a jumper wire at the modem end of the cable connecting it with the SOL's serial port. The jumper goes from pin 4 to pin 23. The modem needs high or low voltage on pin 23 to change baud rates. Pin 4 of the SOL serial port can provide it. Pin 4 was just sitting there doing nothing like the original relay.

That takes care of all the hardware changes. They really aren't very difficult. What was difficult for me was figuring out what to do since all this hardware and electronics stuff was new to me when I first got my SOL.

All that remains is the software to make it all happen. As I mentioned, the SOL will come on at 1200 baud. To switch the modem to 1200 baud, your communications program will need the following instructions in the initialization portion:

```
        MVI A,MHI         get high speed bit
        OUT MST           switch modem
```

These instructions switch both the SOL and the modem to low speed:

```
        MVI A,SLO         get low speed bit
        OUT SBD           switch computer
        MVI A,MLO         get low speed bit
        OUT MST           switch modem
```

To switch everything back to high speed, use the following:

```
        MVI A,SHI         get high speed bit
        OUT SBD          switch computer
        MVI A,MHI        get high speed bit
        OUT MST          switch modem
```

The equates used for the above are:

```
MST EOU 0F8H             Modem STatus port
SBD EQU 0FAH             Speed relay port
SHI EQU 00H              Sol HI speed bit
SLO EQU 40H              Sol LO speed bit
MHI EQU 10H              Modem HI speed bit
MLO EQU 00H              Modem LO speed bit
```

For a communications program, I am using a modified version of the Micro Communications Package that appeared in the March 1980 issue of Interface Age. It took a few months to get that working because I didn't know anything about assembly language programming when I started. The original program helped in the learning process. I thought it was very well structured and well commented. That made it easier to eventually understand.

First, I made the program, as it appeared in the magazine, work with the Sol/Helios. After I typed in all the source code, the first time I assembled it there were over 100 errors flagged. Eventually I got it to work. This is where I learned to use the Debugger. Then I added routines to read from and write to disk files. After that, I put in the baud rate switching. At that point, I couldn't remember all the commands to change speeds and duplex mode, etc. (about 10 commands total). So, I put in a routine to print a menu on the screen.

I'd like to share the program with Proteus members. I'll contact Interface Age and ask permission.

In using the communications program, the information received goes into a regular PTDOS file. Most of the time I then copy it into Word Wizard to print it out. I also used WordWizard to print out assembly language listings. This way the printing pauses at the end of each 8 1/2 x 11 page while I put in more paper (I need a forms tractor). One problem I encountered was that WordWizard space averaging messed up the format of the listing or information received on the modem. A cure is to add the WordWizard return symbol (control A) at the end of each line. This gets to be laborious if you do it line by line to a long file. I have discovered that you can use the WIZ command on the WordWizard systems disk to do it for you. The WIZ command is more than just an Electric Pencil Sharpener. Lately I've been using it quite a bit. It could be that most WordWizard users don't even know it's there.

The procedure is to start in PTDOS with the file to be WIZed on a disk in slot 1. Then remove the PTDOS disk from slot 0 and insert the WordWizard systems disk. Don't bootload WordWizard yet. While still in PTDOS type:

```
     WIZ filename/1,nn <CR>
```

where nn is the line length you want the file to have in WordWizard. If I want to preserve the original line length of the PTDOS file, I use a big number like 100 for nn. It doesn't matter if nn is bigger than the original length. WIZ copies the PTDOS file into ARCH on the WordWizard systems disk and adds the control-A carriage return to the end of each line. When it's done WIZing, remove the disk from slot 1 and insert a Document disk. Then type BOOTLOAD to get into WordWizard and use the Retrieve command to copy the file from ARCH onto the Document disk.

Add Computer Port to the list of people still supporting Sol/Helios. They've always been very helpful when I've called them with a question. They're even still selling Sols. They also still have some Processor Tech software. I just bought PTC Fortran from them. We are all fortunate that they continue to provide this support and still writing additional software for us to use on our Sols.

Their new address is:

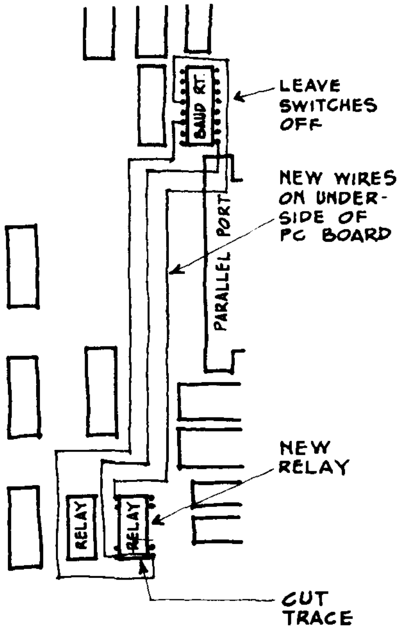
Computer Port
2142 North Collins
Arlington, Texas 76011

This is starting to get lengthy. I hope the information is useful to other Proteus members. How about an update on what (if anything) Proteus is doing with SLAC Pascal. Will we ever be able to use real numbers?

Cordially,

Mike

Michael A. McKelvey



random access

SOL SYSTEM IV FOR SALE

Sol System IV, 64K memory, Helios II, dual 8 inch floppy disks, TV monitor. With circulation-advertising software, miscellaneous disks. About 4 years of lease remaining. The Eastern Oregon Review, P.O. Box s, La Grande, OR 97850. (503) 963-5432. Herbert E. Swett; Editor and Publisner.

LOOKING FOR I-CHING

I have a Sol-20 and Mr. Gary Ingram (president of Processor Technology) had a program for the Sol called "The I Ching". I would very much like to obtain a copy of it. If anyone has a copy they can call me collect. C.E. Goodson, 3128 Edgehill Rd, Fort Worth, Texas 76116. (817) 732-5469

LOOKING FOR VOLUNTEER HELIOS SYSTEM

I'm looking for a volunteer who has a HELIOS system in the Los Angeles, Orange area of California. We're trying to make a memory board work with the Helios Disk Drive. Beta Computing Devices of Orange, Ca. have a logic anaylizer and we need a Sol/Helios computer system brought to the store for taking timing pulses. If interested call Larry Hunley (714) 633-7280 P.S. They have a nice 64K memory board priced at only $595. Frank Heyer, 40 W. 3rd street, Freeport, New York 11520

NEEDS FIX FOR G/2 SOL EXTENDED BASIC

I, like many purchasers of G/2 Sol Extended Basic from GRT corporation am left high and dry because the fix of the "bug" in the save command for saving programs on cassette was not forthcoming as promised. If any of your readers has a fix for this "bug" it would be greatly appreceiated, and would be communicated to others who have the same problem. Joseph Freal, 22040 S.W. 164th Avenue., Goulds, Florida 33170.
(Editors note: Please send it also to PROTEUS so we can publish it!)

WANTS TO INTEGRATE Z-80 WITH THE SOL

I am a Sol owner who intends on keeping his machine for lack of a more cost-effective one. I would like to find out any information or names of others who have successfully integrated a Z-80 into their machine and still run the majority of their software. I know of a "Zol" conversion whereby a Z-80 CPU may be plugged into the backplane, but is there anytning (like the Dutronics conversion for IMSAI boards) which works for the Sol? (IMSAI used standard INTEL support chips but Processor Technology didn't.) I'd like to use my Z-80 macro assembler but can't.

I have looked at Z-80 CPU boards and realize the pin and signal incompatibilities (some signals must be faked) but the Sol with its multiplexed internal bus presents unusual difficulties here.

Thanks for any information you may have. Gordon Wong, 101-2121 W 6th Avenue, Vancouver, B.C., Canada V6k 1V5

Dear Tony:

First a personal comment: Don't spread yourself too thin; you will last longer that way. This is purely selfish on my part since I regard your efforts for Proteus highly.

Secondly, I would like to recommend two products to the membership:

1. You can believe the Neutronics advertisement claims for their 64K RAM, JAWS. It is a remarkably uncluttered, cool-running, well-designed board. My 48K version ($449.95, delivery within 2 weeks) runs solidly (no flickers, drops, or crashes) in my Sol/N* system. What a pleasure after using two PTC 16K boards for 2 years! I don't know about its behavior in DMA systems. (Personal observation: it is great to use hardware from the East coast for a change:- everything else in my system is from the West.)

2. "Tiny" Pascal (Chung/Yuen) is available for N* and CP/M from Supersoft: fast, elegant, with source as well!

Lastly, I came across a statement in the first column on page 18 of issue #6 (June 15, 1980) of Computer Shopper that Sol is going back into production! Whats the story?

Sincerely,

Ron

Rinaldo F. Prisco

VISTA 200 DISK SYSTEM REVIEW

I just received the Vol 3,#2 issue of PROTEUS/NEWS and noticed the request by Bob Freeman for information about the VISTA V-200 Disk Svstem. I have two Sol-20's and last fall equiped them with these disk systems. Both Sol's have 48K of memory and I would like to comment on the memory also. The V-200 consists of a controller board, one or more Micro Peripheral Inc. B-51 double density single sided drives, and the usual CP/M software on disk. The particular beauty of the VISTA system is that a customized BIOS (called SOLBIOS!) is provided which believe it or not means that a Sol owner who wnats a disk system now can actually buy a "turnkey" one. I can testify that you literally can unpack, plug in, turn on, type EX D900 (CR) and be in CP/M!

My memory boards consist of PTC 16KRA and 32KRA, a WAMECO MEM2 16K oard populated with 2114 chips and an SD Sales LExpandoram board populated with 32K or 4115 chips. Anyone reading back issues of PROTEUS would conclude that I should be having plenty of memory problems. The two PTC boards were repaired by them in the spring of 1979 just before they folded and I have had no troubles with them since. The other two boards worked when first turned on and continue to do so even though others have reported difficulties with them particularly with disk systems.

The performance of the V-200 systms has been flawless. Since their installation in October, 1979 there has been no down time. The SolBIOS contains drivers for disk I/O and Solos CRT and keyboard I/O. Skeleton drivers for other I/O (printers, etc) are provided which must be modified for the additional peripherals.

If any PROTEUS members are interested in this disk system I suggest they contact Mr. Gary Cassidy of VISTA Computers, 1401 Borchard, Santa Ana, Ca, 92705. When I last spoke with Mr. Cassidy he told me that they now sell their VISTA -200 systems with any of the Mirco Peripherals drives. Thus you can have from 400K bytes to 1.6Mbytes on a dual drive system depending on whether you use the B51, B52, B91 or B92 MPI drives.

Charles H. Stembridge

SOL BASKET-CASE NEEDS ASSISTANCE

I just aquired a Sol-20 (basket-case, but that's the only kind I could afford) and a friend told me about PROTEUS(SOLUS). It had power supply problems which are now solved, but it still has a few little quirks.

1. 50-75% of the time when it is turned on, the bottom 1/3rd of the screen if filled ,with apparantly random characters, A keyboard reset always clears this and results int he the SOLOS prompt.

2. Occassionally (4 times last month) I have had trouble with the cassette I/O. One those occasions both GET and CAT will start the cassette recorder for several seconds then it will stop. This may happen several times before the system will work, and it can happen with either cassette recorder, different cassettes (even blank ones).

3. One of my memory boards is an IMSAI 32K RAMIII. Sometimes when I load TREK80 on this board it won't run. The initial display comes on, but the face is not there and the cursor does not move(it is stuck under 6), and the game never begins.

If I run the ESV Diagnostic RAM Test the memory board checks out ok and TREK80 will then load and run without any trouble!

Any ideas or hints you can give me?

John Nickel

MEMORY FILL TEST AND REMARKS ABOUT "FOR SALE" ADS.

I am writing To tell you that I've not yet received the Mar/Apr issue of PROTEUS. I realize that the problem may be because I had second thoughts about parting with $18 for such a slim magazine, but the thought of being a lone computer enthusiast finally won out.

I would like to see more HARDWARE AND PROGRAMS in the magazine and LESS "For Sale" Ads. These are nice, but not every issue.(Maybe once or twice a year.)

I have only had my Sol-20 for approximately one year and so far have gotten only 3 Processor Technology's "ACCESS'S" I treasure them! Infact, I would like to borrow a complete set to photo-copy...any leads?

I am also enclosing a very short memory fill test(16K bytes) that works beautifully. It was shown to me by Steve Miller of Moline, Ill.

I load it at 0000(H) and it will fill the remaining memory with whatever and then dump to the screen to check for errors.

```
0000  21
      10 00      lo & hi bytes of start address
0003  3E
0004  00         byte value (use 00,55,AA,&FF)
0005  77 23 7C FE
0009  90         hi byte of last mem location plus 1.
000A  C2 03
000C  00         hi byte of where loaded
000D  C3
000E  04 C0      jump back to Solus
```

1. Enter at 0000(H)
2. EX 0
3. DU 0-8FFF (36K in my case) To change byte value EN: 04 CR and repeat procedure.

I don't know; maybe this is a commonly known code. It has really helped be and I think it may help other novices. I know it isn't on cassette, but is so short it's no problem to type in and verify.

Tom Boerjan
Rock Falls, Il

P.S. I have C1 and there is one program "FINAN" by Keith Turner that I can't figure out. Can you give me his address or any help?

(Editors Note: About the $18.00 cost for "such a slim magazine": As you can see, we do not solicit advertising from major companies. If we did sell space then the newsletter would be twice as large but have the same amount of information. The issues vary from 20 to 24 pages. Now, you will note that the print seems smaller than usual. This is due to the fact that we place on EACH PAGE three pages of information and then have it shrunk by the printer to fit on regular paper. This means that a 20 page issue is really 60 pages of information and a 24 page issue is really 72 pages of information. You could subscribe to BYTE or INTERFACE AGE and for $15-$18 you would get two to three Sol related articles to read and enjoy. I feel that $18 for six issues of this newletter is a bargan!

About the more HARDWARE & PROGRAMS rather than "For Sale" Ads: If we would receive more Hardware and Program articles I would print them. As you can see, in this issue, Fr. McGahee is a constant contributer and the problem lies with the fact that except for him and several others; they are the only major contributers. Please, if you have any articles, no matter how dumb, trite or whatever, send them in. If you cannot type them on a sheet of paper, then I will type them for you. As for the "For Sale" ads all I can say is that it would be nice if the prime forces of economics were such that someone would only need to buy or sell something once or twice a year. I have found that most people who need something or those who need to sell something appreciate the Want Ads and they also are catered to by this newsletter. In many cases items come up for sale for a limited time only. A good example was the Processor Technology bare boards I was selling for a third party. Someone invested a lot of money in buying several hundred sets of bare Proc.Tech. boards and asked me to sell them for them. The problem was that there wasn't much time to get them sold. The person wanted to unload them as soon as possible and also left it open to other buyers. If I or PROTEUS had the money it might have been possible to buy them ourselves and keep them for you but I just didn't have the money. So, by the time I was able to get the word out in the newsletter and by phone to those who asked to be called when such a buy was available, 90% of the boards were not available anymore. I had to sell them on a first call, first served. Out of this sale over 40 of PROTEUS members were able to get boards that were sold at a price that will never occur again. They are happy and I am glad that this newsletter was able to help them.

Sorry, if I sound like complaining because I wouldn't be doing this if I didn't enjoy it. If this were a "professional" magazine you wouldn't get the personalized service. We may be slow but we care!)

REGARDING LEE FELSONSTEIN

I just received my April/May issue of Proteus/News and fund the information on page 19 about the work Lee Felsenstein has mapped out very interesting, particularly about relocating Solus to upper memory...keep us informed! Also, has anyone adapted Extended Cassette Basic to a NorthStar system? And could you give us more information on how to move (relocate) NorthStar D.O.S.? All in all if you guys keep up the good work, who needs Processor Technology?

Jeffery W. Nelson
Brooklyn, N.Y.

GAMES WANTED

Hayden Publishing Company, Inc., the publisher of SARGON II, is once again accepting sophisticated and interesting new games for ALL home computers.

If you have a game which you would like us to consider for national and international publication, please contact:

Stephen Radosh
Games Editor
Hayden Publishing Co., Inc.
50 Essex St.
Rochelle Park, N.J. 07662

We are also looking for people to add hi-res graphics to existing game programs, on a free-lance basis. Please contact Mr. Radosh at the above address.

HAM RADIO APPLICATIONS

Dear Stan:

I have been meaning to write to you for quite some time now, regarding a variety of matters, so please let me get right to them.

First of all, I do owe you an apology. Late last summer you sent me some literature that you had picked up at a computer faire, concerning Ham Radio Applications. At that time, you asked if I might look over the literature, perhaps borrow a piece of the equipment from the manufacturer, and write a review for Proteus News. Stan, at that time, I found myself working out of town for a lengthy period of time, and just could not honour your request.

What I did intend to do however, was to attempt to write an Assembly Language Program to use with the Sol to receive and decipher Morse Code. The program would have been used with a simple audio to digital converter which would hook up to the Sol Serial Port. Now for me, this would be a task equivalent to an inexperienced person successfully climbing Mount Everest in a snowstorm. I thought, however, that this would not only be a tremendous learning experience for me, but if successful, would produce a very valuable piece of software, not just for Ham radio operators, but for all users of Sol.

What happened then you ask? Well, shortly after beginning my project, I had trouble with my 16KRA memory board. It turned out to be a burned out Data Delay unit....that little devil that sits in the lower right hand corner of the board. The company that I deal with in Toronto had difficulty in locating the company that supplies these units, and to this day, I still do not have one, although this is the week that I have been promised one in the mail. Here's the real Zinger.... I've been told the little unit will cost me between $30 and $50. Have you ever heard of such a rip-off??? Apparently the supply company is in New Jersey somewhere. Have other members had this problem. I tried to simulate the Data Delay with several outboard circuits that I built up, but with no success. Has anyone of your clever hardware types been able to do this Stan. That would make a tremendous article. Anyway, when I get back in business, and if I have any success with my Morse Code project, I'll let you know.

Now some other things. You have stated several times in the pages of Proteus, that you will not abandon the hobby types who you originally appealed to in the first place. Well Stan, the last two issues of Proteus from last year seemed to contain fewer letters, articles of interest, and programs for the hobby-type user than has been your standard. In fact, I feel that 50% of those last two issues made up a catalogue of Software for the Sol. Now don't get me wrong, I feel that a catalogue of this type is invaluable to a Sol owner, but I would rather see it published as a separate piece of material, and keep Proteus as the great information and software publication that it was. Have others questioned you about this too?

Nevertheless, I am sending in my money to re-order Proteus for another year. This you will receive under separate cover Stan, so that I can ramble on a little more here in this letter.

Am I too late to take advantage of the opportunity to purchase Source Code for Extended Cassette Basic? I am very interested in this, and would appreciate any information that you could send me in regard to the matter Stan.I am also interested in Source Code for ALS-8 revision B, and TREK-80 and Gamepak.

Another question Stan, I have often wondered if you received more information and articles from Proteus members each month, whether you could and would make each issue bigger, with more pages and articles. Is it simply a matter of just a few carrying the load, or do you have to keep the size of Proteus down to keep your costs down??? If you would like me to write an article to encourage members to write in, no matter how trivial they think their information might be, just let me know...I can certainly do that for you.

Hey, what about a group project....such as the members of Proteus getting together and writing our own word processor, etc. This could be an on-going project, with one portion of the entire program published in Proteus in each issue until the complete program is complete. I would suggest, of course, that this be done in Assembly Language. First determine what features would be required, and then break them down into separate programs. Run a contest each Month to see who can write the program in the most efficient way, and then incorporate, his entry as part of the completed project. Just a thought Stan!

Several issues back, someone asked how we could read Radio Shack TRS-80 tapes into a SOL and thereby have access to a lot of great software. There has been no follow-up on this, and I think personally, this should be a top priority project of Proteus. Some of the software already in Proteus has amazed me, in the cleverness of it's writer and so I am sure that someone out there has the knowledge to do it. By the way Stan, I have learned what Assembly Language I know by studying the various routines in Proteus, along with reading a few other text books that I have. It is in this regard that I find Proteus invaluable, and so would hope that you intend to keep it a publication of this type, rather than something else.

Thanks for taking the time to read this Stan. My questions above about the Source Code are of Prime Importance to me right now, so a reply would be much appreciated.

Sincerely,

Larry Michael Mandzuik.

Answer to Larry's letter:

Larry,
Your comments are well taken. The Software Directory was something I promised to do long ago, and finally got it completed so I wanted to get it out to answer many questions I had received. If we do it again it will be as a separate item.

We receive an adequate flow of material, but its not always what appeals to the majority. Cost does limit the size of the issues.

About the Source Code: Its not to late to get in on the purchase, but we don't yet have the goods to sell. PTC has been held up in delivering it to me because of a law suit. It is supposed to be ready to release around the end of March.

Stan

(Editors note: Stan has been doing his best trying to keep on top of the Source Code buy. As you must understand, a deal like this takes considerable patience, especially when dealing with lawyers, etc. Regarding the group projects you mentioned I can only say if someone stands up and says "I am willing to work on something" then I will be glad to publish it and maybe someone else will contact you to assist. I know for my self that I just do not have the time to give any other projects. I receive on the average of 30 calls a week regarding SOL users. Answering their questions and doing what other projects I do more than takes up all my spare time.)

WANT TO BUY

One or two E revision working condition Sol-20' without memory. Willing to pay $500.00 each, including parcel post, and each Sol-20 mailed in two boxes, one with the CPU, backplane, keyboard and manuals, the other box should contain the POwer supply and the rest of the SOL-20.

George G. Warner KEG
American Embassy Bangkok
San Francisco APO
96346

FOR SALE: Sol-20 partially assembled. $800 or best offer.

Clarence Wickers
2301 Sycamore Drive #11
Antioch, California 94509
work phone (2:30-11:00 p.m.) 398-3443 (area code 415)
home phone (415) 754-8914

Dear Stan,

I just finished reading the Jan/Feb issue of Proteus and I want to comment on several things. First a big THANK YOU to Tony Severa whom I assume is the new editor. I really liked the editorial, especially the part about making money with your computer. Personally, if I'm ever to have disks, this hobby is going to have to pay for itself.

Second, many thanks to Bob Werner for sharing his fix for those $#%&!! REM statements in PTC Extended Cassette Basic. I must admit that I haven't had time to try Bob's fix yet but will as soon as I can.

Third, thanks also to Ed Bolton for his generous Basic modification to PTC ECBASIC that eliminates LETS and blanks from the exponents of numbers when writing to tape. While I don't plan to use this now, it will be very useful.

Fourth, I need help! Does anyone have a program that will produce a cross reference list of the variables used in a BASIC program. I've seen this only in I.B.M. Coursewriter BASIC: you just type in XREF and it prints a list of the variables in alpha sequence and the line numbers each is used in, just like the one produced by ALS-8 when you assemble a program. I would be very grateful to anyone who has done this for PTC ECBASIC. If nobody else has written a x-ref function I am going to attempt to write a program in Basic that will read a text file version of a Basic program and generate a cross reference table.

Fifth, everyone please keep up the good work!

Best regards,

Robert W. Heerdink

MATRIX OPERATIONS

One of the most powerful features of PT ECBASIC is the group of MAT statements. Many non-mathematical persons are not aware of how useful they can be in speeding up business-type programs. As an example the user’s manual on page 7-2 has a short program that adds two arrays using FOR-NEXT loops. Using 30*30 arrays the running time is 14.6 seconds; substituting MAT Z=X+Y for the loops cuts the time to 1.2 seconds. Try to match that with an Apple or TRS-80!

The accompanying program demonstrates a practical application; keeping running totals by column and by row in the manner of Visicalc. The demo is for visual display so the amount of data is limited but printed reports of any size can be produced by altering the array dimensions. Possible applications are endless, budgets, statistical reports, schedules, and project cost forecasts,etc. Set up one matrix for income and another for planned expenses, subtract the two and you’ve got a cash flow plan for the entire year. I then substitute actuals at the end of the month to update the plan.

I learned the technique in a course taught by Gene Barnett. His text, listed in the BASIC5 manual bibliography,is one of my favorites. It’s crammed with clever programming techniques.

LEN KALISH

EDITOR'S NOTE

As I mentioned in the last newsletter, I have taken control of Proteus News once again. This newsletter was partially done by our former editor, Tony Severa. I've been overwhelmed by business demands on my time so we're way behind schedule. I'll try to put out a thicker issue next time to get caught up. I have several people lined up to help me distribute the PTC source code; I just haven't been able to get together with them yet. Soon though. I've located ALS-8 source code, finally. More in the next issue.

Stan

```
LIST

  10 REM.                   *** CALC ***         JULY 1980
  20 REM.                 BY LEONARD KALISH
  30 REM.             580 S. SAN VICENTE BLVD. #3
  40 REM.                LOS ANGELES CA,90048
  50 REM
  60 REM.      CALC DEMONSTRATES THE POWER OF SOL'S MATRIX
  70 REM.      OPERATIONS.  BY ELIMINATING LOOPING,PROGRAM
  80 REM.      RUNNING TIME IS VERY MUCH FASTER.
  90 REM
 100 REM.      CALC IS ALSO A USEFUL TOOL FOR DEVELOPING AND
 110 REM.      REVISING BUDGETS AND SCHEDULES; YOU CAN SEE
 120 REM.      IMMEDIATELY THE EFFECT OF CHANGING ANY ELEMENT
 130 REM.      IN THE PLAN.
 140 REM
 160 DIM A(11,6),T(1,6),S(6,1),S1(11,1),S3(1,1)
 170 DIM A2(6,11),T1(11,1),T2(6,1)
 180 REM.FILL WITH ZEROS TO INITIALIZE (OR USE STORED DATA)
 190 MAT A=ZER
 200 REM
 210 REM.      FIRST,LETS CALCULATE THE TOTAL OF EACH ROW
 220 REM
 230 MAT S=CON
 240 MAT S1=A*S
 250 REM
 260 REM.      SUMMING THE COLUMNS USES THE SAME TRICK BUT
 270 REM.      REQUIRES SWAPPING COLUMNS AND ROWS IN 2 PLACES
 280 REM
 290 MAT A2=TRN(A)
 300 MAT T1=CON
 310 MAT T2=A2*T1
 320 MAT T=TRN(T2)
 330 REM
 340 REM.      FINALLY, GETTING THE GRAND TOTAL IS EASY; JUST
 350 REM.      WATCH THOSE ARRAY DIMENSIONS!!
 360 REM.
 370 MAT S3=T*S
 380 REM
 390 PRINT "R\C";
 400 FOR C=1 TO 6: PRINT %#8I;C;: NEXT : PRINT "   TOTAL"
 410 FOR R=1 TO 63: PRINT "-";: NEXT
 420 FOR J=1 TO 11
 430   PRINT %31;J;
 440   FOR K=1 TO 6
 450     PRINT A(J,K);
 460   NEXT K
 470   PRINT S1(J,1)
 480 NEXT J
 490 FOR R=1 TO 63: PRINT "-";: NEXT
 500 PRINT "SUM";
 510 FOR L=1 TO 6
 520   PRINT T(1,L);
 530 NEXT L
 540 PRINT S3(1,1)
 550 INPUT "ENTER ROW,COLUMN,AND VALUE:",R,C,V
 560 LET A(R,C)=V
 570 PRINT "&K"
 580 GOTO 200
```

ORGANIZING CASETTE FILES
Leonard Morgenstern

I have been using this system for organizing tape files for over 18 months, and find that it works quite well. Instead of using "END" to mark the end of a file or group of files, I use a "DATEMARK". This is a 5-character tape label in the following format: AYMDD, where A is some alpha character (normally the letter "D"), Y is the year (0=1980, 1=1981, etc.), M is the month in HEX (1=January, 2=February, A=October, B=November, C=December), and DD is the day of the month). Thus,

```
SAVE D0322 0 FFFF
```

would save an empty file (label only). As an adjunct to this system, I use a diary as a log book, and enter the date along with a description of what was stored on the tape. This is very useful when you are faced with a tape whose existence you may have entirely forgotten.

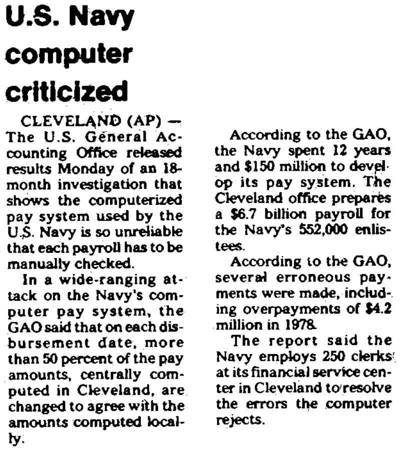
The use of the addresses 0 FFFF to write an empty file is not as well known as as it should be. If you are storing files from PT-EDIT, the command)D0322$OPE will have the same effect.

Berry's World

"Golly! Just a minute! I'll check my home computer to see if there is any more beer in the refrigerator."

What do you do with a home computer anyway?

...And you think you have problems? Look at this...



U.S. Navy computer criticized

CLEVELAND (AP) — The U.S. General Accounting Office released results Monday of an 18-month investigation that shows the computerized pay system used by the U.S. Navy is so unreliable that each payroll has to be manually checked.

In a wide-ranging attack on the Navy's computer pay system, the GAO said that on each disbursement date, more than 50 percent of the pay amounts, centrally computed in Cleveland, are changed to agree with the amounts computed locally.

According to the GAO, the Navy spent 12 years and $150 million to develop its pay system. The Cleveland office prepares a $6.7 billion payroll for the Navy's 552,000 enlistees.

According to the GAO, several erroneous payments were made, including overpayments of $4.2 million in 1978.

The report said the Navy employs 250 clerks at its financial service center in Cleveland to resolve the errors the computer rejects.

Syntax Corporation
4500 W. 72nd Terrace
Prairie Village, Kansas 66208
Phone (913) 362-~~7043~~ 9667

June 30, 1980

Stan Sokolow
Editor
Proteus/News
1690 Woodside Rd, Suite 219
Redwood City, CA 94061

Dear Mr. Sokolow:

Financial Consultants have recently taken an active interest in acquiring a computer system to help them in their business. But the consultant needs a problem solving computer rather than a bookkeeping computer.

The June, 1980 issue of the Financial Systems Report (enclosed) discusses the problem of "Business Versus Personal Computers" as it applies to financial & tax consultants.

Some of your readers may be concerned with this problem. I would be glad to send a free copy of this issue of the Financial Systems Report to any of your readers.

Requests for a free copy should be sent to V.K. Jacobs, Editor, The Financial Systems Report, Box 8137, Prairie Village, KS 66208.

In addition to the June issue of FSR, I am enclosing the May, 1980 issue which discusses the availability of financial forecasting and modeling programs. If you would be interested, you may reprint that article in "Proteus/News" - or any other article in these two issues of the Financial Systems Report.

Yours truly;
Vernon K Jacobs
Vernon K. Jacobs
Editor & Publisher
The Financial Systems Report

Encl.

[If you are interested in financial planning, write to these people for a sample newsletter. - Stan]

CP/M USERS GAIN IMPROVED KEYED FILE SUPPORT

Van Nuys, Calif. - Micro Applications Group, a microcomputer systems house here, has introduced MAGSAM IVtm, a new high-performance version of their keyed file management system. This enhanced system combines the features and capabilites of MAGSAM III with the speed and performance of 8080 assembler. The result is access times are reduced up to 75% compared to previous versions of MAGSAM.

MAGSAM IV enables system developers to create programs that access data records quickly and directly by user defined keys. Secondary indexing with any number of keys is provided to allow access to data by any and all desired data elements. Real-time record and key deletion with automatic reclamation of free space conserves disk space while simplifying program development.

Record retrieval techniques provided with MAGSAM IV include random by key, sequential by key, generic by key (wild card search), sequential in physical order, and random by record number. Records may be created randomly by key and sequentially by key, and updated by any of the retrieval methods. Key and record deletes may be performed randomly by key.

MAGSAM IV is provided with a subroutine to interface directly with CBASIC programs. This allows MAGSAM IV to be intitiated by simple GOSUB statements and maintains compatibility with previous versions of MAGSAM. MAGSAM IV requires 8K of memory over that occupied by the operating system and the calling program.

Each MAGSAM IV package includes the MAGSAM file manager in pre-loaded and relocatable object code, MAGSAMX tutorial program, MAGSAMD file dump utility, User Guide, Reference Card, and one year update service. The 100 page User Guide provides a description of the general principles and applications of keyed file structures as well as detailed reference and tutorial information on MAGSAM.

MAGSAM IV is available on standard 8" and Micropolis Mod II diskette formats. A single site license for the MAGSAM IV package is $295. The User Guide is $25 separately. All products are available from Micro Applications Group, 7300 Caldus Avenue, CA 91406.

****

MAIL ORDER & MEDIA ANALYSIS SOFTWARE FOR SOL

MAGAZINE ADVERTISING RATE CARD ANALYSIS: DISK VERSION
Requires 48K Sol, North Star format CP/M, two s.d. disk units.

Rapid production of in-depth analysis of advertising space rates. Files rates for future reference, comparisons. Applications: structure rates for profit, compare with competition, combine with audience studies to derive costs to reach a market segment. Analyses: cost per thousand, unit frequency and continuity discounts, fractional page imposts and premiums, incremental discounts. Compare publications A and B for variances in rate and CPM. Three days manual work reduced to 3 hours. Fast-running, listable MBASIC code, and user documentation included.

PRICE:$600.

(A sample disk of 7 representative rate analyses, without filing capability, is available for 32K Sol disk system in North Star-CP/M CBASIC2. Also for 16K Radio Shack TRS-80 Level II, tape or disk. Sampler price: $100.)

****

MAIL ORDER BUSINESS PROFIT ANALYSIS: DISK VERSION
Requires 32K Sol with North Star-CP/M.

Analyzes profitability of direct mail selling efforts, given information on response, product and shipping costs, size of mailing, marketing expense. Output includes unit marginal cost, gross profit, net contribution to profit and overhead, inventory valuation and cash flow. Subprogram will increment estimated sales reponse to produce "range" forecasts and pinpoint breakeven sales levels. CBASIC2 run-time package, user documentation included.

PRICE: $100.

DANIEL S. HUNT / PUBLISHING STRATEGIES / 16 Escapade Court / Newport Beach, CA 92663

EDITOR: FOR THOSE WHO MISSED IT!

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

BUSINESS

Well, your a business person and you have decided to look into the possibility of buying, leasing or using computer services in some aspect of your business. You probably don't have a background in computer science and been keeping up to date with the latest machines and developments in business computers. That's why you have come to me. I am a Micro-Computer Consultant. As a consultant I am available to you to provide you with what you need. It can be as little as information on types, costs and capabilities to as much as assisting you in setting up and operating a system of your own.

When business people think about computers several questions may come to their mind:

1. Do I really need a computer in my business?
2. Will Micro-computers be capable of satisfying my needs now and in the future?
3. What kind of computer should I get and what are the hidden costs for support equipment?
4. Will the system I get be utilized efficiently?
5. Will the system I get be expandable?
6. Who will repair the equipment and what kind of maintenance cycles to they require?
7. How do I get the most for the least?
8. What kind of training will I and my employees need to be able to operate the computer?

My purpose is to help you find the answers to all the above and other questions you may have before you have to risk the investment of your money. There are over 20 periodical journals, trade magazines, user journals and books being released every montn that addresses itself to micro-computing. A large part of my job is reading and keeping up with all the information that is coming from the field. The Micro-Computer field is relatively new (only 5 years) but the amount of information being produced has been explosive.

To be able to assist you to the maximum we need to speak the same language and this paper is designed to assist you in understanding the words I will be using in our discussions. Please feel free to circle any part that you do not understand and we can discuss it when we get together.

HISTORY

The beginning of computers as we know them can be traced to a French mathematician, Blaise Pascal, in 1642. It was made of wheels which when rotated was able to function similiar to the adding machine of today.

Charles Babbage, of England, designed a machine called the "difference engine", that was able to calculate values of a polynomial in 1812.

Babbages ideas continued to the 1940's when the first large general purpose digital computer was built by I.B.M. and Dr. Howard Aiken of Harvard University. It was electromechanical in nature and it was called the Mark I.

Then ENIAC (Electronic Numerical Integrator and Calculator) was built in 1946. It was the first that utilized only electronic (vacuum tubes) parts.

By 1951 the UNIVAC I was being produced for the open market. Since 1951 the development of computers has continued to include the transistor and the semiconductor. These changes made the computer systems more reliable and faster.

For example, ENIAC would only run for 2-4 hours before a breakdown would occur. The temperature produced by all the vacuum tubes would ultimately take their toll and tubes would fail at a tremendous rate. Maintenance would take 12-18 hours and then the machine would be ready for another 2 to 3 hours of maintenance free operation. It took the basement of a large building and the airconditioner needed to cool it was just as large. It consummed large amounts of energy and it was limited in the kinds of work it could do.

Now, we have computers with more power, more speed, better maintenance times, more up time, no airconditioners needed except for extreem cases and only a need for a small amount of energy to do its work.

One of the myths about computers is that they are mathematical machines and unless you know mathematics and calculus you won't be able to understand them. This is just not true! COMPUTERS ARE SYMBOL MANIPULATING DEVICES. Anything that can be symbolized can be worked with on a computer. For example:

1. The symbols on sheet music REPRESENT music and are not otherwise connected to music.
2. Our language is symbolized by words. Words have no other meaning other than that which we give them.
3. Our language REPRESENTS our ideas and thoughts.
4. Art can symbolize sound, ideas, abstract concepts, feelings, and emotions.

Because the mathematicians got to the computers first we have to put up with their "buzz words". If you want to be a programmer then at least some concept of algebra is needed; but to be able to use one and to understand what it does for you, it is only necessary to compare the computer with our body.

For example: The CPU (central processing unit), in the case of micro-computers is the decision making part of the computer.

For us to be able to be aware of our surroundings, thoughts, feelings, etc. we need memory; so does the computer. We call that memory RAM (Random Access Memory). It is the part of the computer that is similiar to our awake memory. It is a sort of scratch pad where all the conscious decisions are made. It is short term memory in that if the information in it is not refreshed it will forget what it knows.

Since we don't want to spend our awake time thinking about our heart beats and breathing we have an autonomic nervous system. It controls our body's temperature, blood pressure, heart rate and breathing so that our conscious memory doesn't need to spend valuable time and energy doing it. The computer uses a memory called ROM (Read Only Memory). It is commands that are fixed in memory that the computer knows is there and can call upon it as needed.

Well, now we have the CPU (decision maker), the RAM (short term awake memory), the ROM (automatic memory) we need to utilize something to make it operate. For our brain and memory we use blood. It brings the energy producing nutrients and oxygen to our brain. In the computer the device that does the same function is called the power supply. It supplies the electricity to the computer to make it operate. Like our blood it is necessary for it to bring the nutrients in the right amounts. It must be able to provide enough energy and exact quantities to the computer if it is to operate.

We now have an operating decision maker with memory to keep all functions operating and memory for decisions we now need something to decide upon. We call this I/O (input/output) in the computer world. In our bodies we have eyes, ears, touch, smell, taste, temperature sensors that connect to our CPU and RAM. The computer needs a keyboard, disc systems and other perpheral devices that do similiar things. They connect the CPU with the outside. To communicate we talk, write, make faces, stand differently, dress differently, etc. To communicate with us the computer usualy prints on paper or on a television screen, but, is not limited to that. Some computers can talk and understand someone else talking to a limited degree. To talk to each other computers can utilize electrical pulses.

kind of research and development costs that they must have incurred? I had been selling their extra Sol equipment so they could meet their payroll. By the time the system was ready to sell the recession hit and they lost several promises from companies to buy. So, what happened to P.T. happened to them. When you cannot pay your payroll, you close your doors, take your licks, and liquidate.

This is not a new story. What concerns me is that so many good people going down the drain with ideas that cannot work without capitol or support from their manufacturers and customers. One of the latest issues that has crossed my desk has been the appearance of cut-rate sales people. I have a friend who, several months ago, fell in love with the new MPI printer. He tried to get a local Computerland store to sell it but they liked the Anadex and PaperTiger better. Since they wouldn't sell it he decided to sell it. He would come to the user group meetings at the store and they would allow him to freely advertise his printer (which he was selling for $100 off the list price!).

But he began to get greedy, so, he decided to not only sell the printer but also other peripherals and equipment, including micro-computers, in direct competition with Computerland. He started advertising his cheap rates in the local newsletters and he was selling his equipment cheaper than the local Computerland was able to. (Many are wondering who is source is) To make matters worst he has now decided to solicit customers right out of the store. The most recent experience I had with him occurred when I wanted to buy a couple of boxes of paper for my MPI printer (I had bought from him.) at the store. While I was talking to the sales person, he comes up to me and wispers so everyone can hear, "Hey, Tony, I know where you can get it cheaper! Hee Hee!"

Well, I bought my paper from the store and not him. In fact, I am not going to buy anything more from him! He does not support me if my equipment goes down! He doesn't stock up on item I might need from time to time! He doesn't offer magazines and books. He doesn't allow me to come in and try a piece of software before buying it. He is so busy selling everyone he doesn't realize the people he is hurting and the possible consequences of his actions.

I am a firm believer in supporting your local dealer! Support them even if it requires paying more for the item. Oh, I know you will eventually buy one or two pieces of equipment from these kinds of sales people, but realize the consequences of your actions. What happens when a computer store goes out of business? What happens to all the people that have come to rely on that business? The cut-rate sales person isn't opening up his own store! He just wants to make a buck!

Think about MicroSun Computers! What's going through the minds of those 25 businesses with Sol/Helios Systems with no maintenance contract? It's so bad, we weren't even able to get a list of the businesses so PROTEUS could contact them and they aren't computer buffs and will probably never come into contact with a copy of a computer magazine. Sound familiar?

Tony Severa
Editor *Emeritus*

PROTEUS/NEWS

Publisher: PROTEUS, 1690 Woodside Road, Suite 219,
Redwood City, California 94061, U.S.A.
Editor: Stan Sokolow, 1690 Woodside Road, Suite 219,
Redwood City, California 94061, U.S.A.
Subscriptions: $18 per calendar year in U.S.A.,
$18 in Canada, $24 elsewhere (U.S. dollars only).
All subscriptions expire at the end of the calendar year (Dec. 31). Mid-year subscribers will receive back issues for current year. Back issues for prior years are sold in annual sets, while available.

CONTENTS

PROTEUS
1690 Woodside Road, Suite 219
Redwood City, California 94061
U.S.A.

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

18104

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

NOVEMBER/DECEMBER 1980 PUBLISHED BIMONTHLY BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA

VOLUME 3, NO. 5 AND 6
DOUBLE ISSUE

EDITOR'S COLUMN
by Stan Sokolow

Finally the holiday season has come and given me a break from my regular work, so I can get down to some concentrated Proteus business. This issue is a double one, to make up for the fact that we are one behind. In the coming year we hope to be able to stay better on schedule, one issue every other month.

I've made some changes to help us do this: we now have a room in my office devoted to the computer and Proteus, soon we will have a phone exclusively for Proteus, and our "operations officer" (Jane Delno) now has space to work when I'm in the office, rather than just on my off-days. I expect that this will allow us to be much more up on things.

Plans for the coming year include publication of my compendium of Processor Tech documentation, that I am calling "Encyclopedia Processor Technica" for want of a better name. I've made arrangements with two very enthusiastic and competent professional programmers who are members of Proteus, to have them be our Helios librarians and to work on modifying PTDOS for more general use on other disks. (See story in this issue.) We have a volunteer who has offered to take charge of maintaining the PTDOS documentation.

Lewis Moseley continues to expand the library of programs available on cassette. Notice Lewis's article in this issue. He now has all of the CP/M user's group library available for distribution on Sol/Cuts cassette. This allows people with incompatible format disks to exchange programs with the library and each other. The interchange program is also available from him to automate the loading and unloading of files to and from the disk.

Our outstanding contributing authors, such as Father McGahee and Joe Maguire, and all of the others who have sent major and minor articles and letters are to be thanked for making Proteus continue to move forward. In spite of the sporadic publication schedule, I continue to get positive feedback from members, and this has kept me plugging away. My role is really just a center for crystallization of the work from all of you.

Best wishes for the New Year,

Stan

Stanley M. Sokolow, Editor

CALL FOR HELP ON LIBRARY DISK H-4

Somehow, in the transition from Tony Severa back to Stan Sokolow, the master copy of Helios library disk H-4 has become messed up. The version we have does not agree with the published Table of Contents in Proteus News, vol. 3, no. 3, page 18. If anyone has an H-4 disk that agrees with the published contents (the "new" corrected version), please send us a copy of the disk. Proteus will reimburse you for the postage and your trouble, and return your disk promptly. Send it to Proteus, Stan Sokolow, 1690 Woodside Road, Suite 219, Redwood City, CA 94061. Thanks in advance, on behalf of the members who want the right H4.

TIME TO RENEW ALL SUBSCRIPTIONS

Once again it is time to renew your subscriptions for the coming year. And once again, I find it necessary to increase the subscription rate. There are some uncertainties in setting our rate annually, since we suspect that there will be a postal rate increase during 1981, and your guess is as good as ours on how much it will be. (An increase in international rates has already been put into effect as of Jan. 1, 1981).

But there are some factors we can predict in estimating our cost. We now have actual office space. (Proteus used to occupy a hall the size of a closet.) I want to get our own phone line. And we have increased printing expenses and salary costs for our secretary.

Also, we used to print many more copies of each issue than we really needed, anticipating that some people will subscribe late, but this has overburdoned us with unsold copies. So, this year we will only print a modest excess each issue. Late subscribers may not be able to receive back issues, but their expiration date will be extended to cover 12 months (6 issues).

Consequently, the 1981 annual subscription will cost $24 for subscribers in the U.S.A., Canada, and Mexico; and $32 in other foreign countries. (Foreign subscribers, please send only U.S. funds.) I grappled with this increase, but to go any less I'm afraid we would have to curtail the work Proteus does. Many of our projects are not self-supporting, such as answering phone calls requesting information, but we feel it is an important part of our service. I still think it is a bargain at the new price, considering the value of just one good tip derived from the experience of other members.

See the renewal form on the outside back cover of this issue.

NEW HELIOS LIBRARY DISK IS AVAILABLE

Library disk H6 is available now. Disk H5 is still in preparation, but it will soon be completed. It will have the source code of a number of device drivers, including the ones from Basic Computer Group for all of the SolPrinters, the Diablo 1610/1620, etc. The drivers will also be in ready-to-use form, so that the non-programmers can simply copy them onto their system disk and be ready to fly.

The contents of H6 are listed below. Due to our increased costs of doing business, we must increase the price of library disks. All Helios library disks will now cost $35 for someone who doesn't donate a library program; or $15 for someone who sends an acceptable program for the Helios library. See volume 3 number 1 issue for details on how to document your programs for the library, but disregard the address given there. Place all orders directly through the Proteus office.

Let me call your attention to two of the gems on H6. MEDIT is a program which acts like PTDOS's EDIT, but it lets you edit files in hex as well as ASCII, and it gives some extra features for searching. It is designed specifically to ease the job of peeking and poking into machine language image files to make changes when you do not have the original source program. RAIDER is the source code for the popular Space Invaders game, reprogrammed to be compatible with Sol or other memory mapped video displays.

Previous library disks are documented in the back issues of Proteus News. Order the library disks directly from Proteus at our newsletter mailing address.

CONTENTS OF H-6 DISKETTE

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

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

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

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

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

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

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

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

MEDIT:D — Documentation for MEDIT

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

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

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

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

SPIN:D — Updated documentation for SPINWR:S.

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

SPIN — A device file assembled from SPINWR:S.

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

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

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

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

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

PROTEUS SOFTWARE LIBRARY NEWS

The Proteus Cassette Software Library has three new features to report. First, we have three new cassette releases, which should be described elsewhere in this issue. Tape C10 has programs of special interest to CP/M users, including a pair of fine programs by Dick Greenlaw which allow the easy transfer of files between CP/M users, regardless of diskette size or format, provided they can read and write CUTS tapes.

Second, we are wasting no time putting Dick's programs to work. Effective immediately, all 42 of the CP/M Users Group Library diskettes are available on tape, along with Dick's tape-to-disk loader program to put them back on disk. The price of these tapes is $10.00 per volume, with no program contribution required. A catalog of the CPMUG volumes is available (also on tape, print it out yourself) for $6.00 through the library. New CPMUG diskettes should be available from us within a few weeks of their release.

Third, we are now offering a media transfer service. If you have an 8" single-density soft-sector diskette containing software that you would like transfered to tape so that you can then load it into your CP/M disk system, we can handle this. Again, the price is $10 for a diskette, regardless of how much or how little is on it. This fee covers the cost of the output cassette, packaging and first class postage. If you want the 8" diskette returned, say so specifically and include return postage in addition to the $10 fee.

Address orders or inquiries to:

PROTEUS CASSETTE SOFTWARE LIBRARY
C/O LEWIS MOSELEY, JR.
2576 GLENDALE COURT, N.E.
CONYERS, GEORGIA 30208

Include a SASE if you want a personal reply or a library catalog.

NEWS RELEASE

CLASSIC PROGRAMS STILL AVAILABLE ON CUTS CASSETTES

Dvorak's Software Review is one of the leading software finding companies specializing in useful and obscure utilities games and user group diskettes. He also has available some of the finest programs ever made available for SOL's. For the most part these programes are otherwise unobtainable.

SOFTWARE MUSIC SYSNTHESIS SYSTEM-an upgrade of the old MUSIC SYSTEM with an additional voice for a Z-80 user. The overall tones are greatly improved and the waveforms and clock speed are user programmable. Requires 8K system and parallel port. No slot required. Also available on Northstar and CP/M diskette...$79.95

ALS-8 (Rev.B). The classic assembler for SOL users. Easy to use. Faster assembly time than most 8080 assemblers. $40.

FASTGAMMON. The Software Review has the license for the VDM version of FASTGAMMON. Without question one of the two or three best games written for the SOL. Machine coded with 100% memory-mapped graphics. A great game and a must for SOL owners. Also available on Northstar diskette. $29.95

All orders please include appropriate taxes and $1.50 shipping and handling. Send check or money order. Order from: The SOFTWARE REVIEW, 704 Solano Ave., Albany, CA 94706

INFORMATION SHEET SMS-1

SOFTWARE MUSIC SYNTHESIS SYSTEM

CONFIGURING SMS FOR YOUR SYSTEM

The hardware interface provided is a multipurpose DAC (digital-to-analog converter) and tempo switch register that can be adapted to any parallel 8 bit I/O port. The board is entirely passive and requires no power supply and does not use a card slot. The board is supplied in kit form with complete and easy to follow instructions. Having only 16 resistors and three capacitors, the board is easily assembled by anyone. An output cable is simply connected to any audio amp with a speaker and an input cable is connected to the computer's parallel port. Because of various user requirements for lengths, these cables are not provided, but are easily obtained from electronic supply stores. To properly jumper the small board for each individual's computer, knowledge of the parallel port output pins is essential, since each computer manufacturer has seen fit to make them all different. Most users easily find the necessary information in their computer handbook or documentation.

Once the hardware is ready the user loads the system into memory and starts the program. At that point the program becomes interactive and begins the configuration process by asking the user questions about the operating environment such as: clock speed, I/O port address, highest memory address, etc. At each step diagnostic tests are made to insure correct system installation.

It is during the configuration process that the user is allowed to modify the four waveforms available. The amplitude and spectral content (up to 15 harmonics) of each wave form are inputted. After each new wave form is inputted, the computer will play a sample scale to demonstrate the tone color. The cycle of modifying and sampling can be repeated until the sound is that desired. The newly configured program can then be saved by the user for future use, eliminating the necessity to have to continually re-configure the program each time the system is used.

It should be noted that the compiler has a limited MACRO capability, and that the CP/M version has two additional programs to convert the SMS programs into the CP./M editor format and back again.

The price of the system on diskette or cassette with complete documentation and with the hardware kit and ten songs is $79.95 (cables not included).

SYSTEM REQUIREMENTS:

Clock Speed: any speed from 1.78-7.10 mHz
Processor: 8080, Z-80, 8085
Operating System: CP/M, IMDOS, CDOS, TSA/OS, Northstar, SOLOS, or CUTER. Eight-bit Parallel output port required (any address). Eight-bit parallel input port used for optional features (any address).
Program size: 4K
Minimum system requirements: 8K
Requires memory with no wait state.
Interrupt driven environment OK.
Program memory residence: CP/M, CUTER, SOLOS at 100 Hex. Northstar DOS at 2D00.

PROGRAM SPECIFICATIONS:

The following program elements are all included in the 4K machine code SMS module:

THE LINE EDITOR: Uses numbered lines to enter and edit the music language source. Any lines may easily be displayed, added or deleted.

THE FILE MANAGEMENT SYSTEM: Used to retrieve or store named source files on cassette tape or floppy diskette.. It can also be used to retrieve, compile and play one or more music files contiguously for performance.

ML/1 HIGH LEVEL MUSIC LANGUAGE COMPILER: The language can accomodate any standard musical notation including: note values between whole notes and 64th notes, triplets, dotted and double-dotted notes, stacatto and pizzicatto, articulation (2 kinds), any key signature, all accidentals (sharp, flat, natural, double sharp, double flat), any time signature, any tempo, any clef (treble, bass, alto, etc.), mixed clef and open score. Any music can be transposed up or down 1½ octaves. If the optional parallel input port is used, the tempo can be altered at any time while the music is playing.

ADVANCED PROGRAMMABLE MUSIC SYNTHESIZER: The heart of the system is the music synthesizer. On a 2 or 3 mHz system it can produce three simultaneous notes. On a 4 or 5 mHz system, it can produce 4 simultaneous notes. Each of the notes (called voices) can play any one of four different tone colors. The standard tone colors built into the system simulate the sound of a trumpet, an oboe, a saxophone and an organ. Any or all of the four tone colors can be altered to produce different sounds. All of the software to change these waveforms is built into the system.

Available from California Software, Box 275, El Cerrito, CA 94530.

(415)527-7730.

News Release

$10,000 GRAND PRIZE

PERSONAL COMPUTING FOR THE HANDICAPPED

(NATIONAL CONTEST)

Johns Hopkins University will be conducting the First National Search/Contest on the Application of Personal Computing to Aid the Handicapped starting in November 1980 and lasting to June 1981.

Major objectives of the contest are to:

a) Focus the power of computing technology on the urgent needs of millions of handicapped citizens.

b) Harness individual innovation & creativity on a national basis.

There will be a $10,000 Grand Prize and one hundred other awards, including 15 personal computer systems for the winning submissions. Entries may be a device, system or a computer program. The contest is open to computer professionals, amateurs and students throughout the United States, with awards in each category.

Categories that may be addressed include computer-based aids for the blind, deaf, and mentally retarded; individuals with learning disabilities, neurological or neuromuscular conditions; and the orthopedically handicapped.

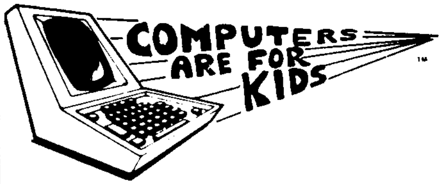
Special meetings on computing for the handicapped are planned at rehabilitation centers and technical society conferences across the country early in 1981.

Grants for this venture were provided by the National Science Foundation and by the Radio Shack Division of Tandy Corporation.

Personal Computing for the Handicapped
Johns Hopkins University
P. O. Box 670
Laurel, Maryland 20810

- **GENERAL CONDITIONS**
  - All invention rights will remain with the contestant.
  - The Johns Hopkins University and the sponsors of the competition will not be responsible for the "security" of inventions.
  - Each entrant must certify to the origination of the invention, i.e., that it is his or her own concept and is not known to infringe any patent or be the rightful property of any other person or organization.
  - Abstracts describing the winning entries will be published in the Competition Proceedings.
  - Entries must conform completely to instructions provided in the 'Entry Information Kit'.
  - The First National Search is limited to residents of the United States.
  - Faculty, students and employees of The Johns Hopkins University and Hospital may participate in the competition but are not eligible for any of the competition prizes.

● Entries must be received by June 30, 1981 to be considered.



4th Annual NATIONAL COMPUTER CAMP

THE ORIGINAL COMPUTER OVERNIGHT CAMP

offering a unique experience in computer education and recreation

SUMMER 1981

For students ages 10-18 with all levels of computer experience including no experience whatsoever. Participants learn to program in Basic and Machine Language. Emphasis is placed on a hands-on approach by providing each participant with ample computer time.

PURPOSE	Educational and recreational computing for all levels of background.
ELIGIBILITY	All students ages 10-18.
ENROLLMENT	One or two week session, two weekers enjoy a continuous two-week program. Small group instruction. Every 2-3 participants have their "own" computer.
DAILY ACTIVITIES	Computer instruction, computer sports tournaments, computer workshops, films, guest speakers, and recreational activities.
RECREATION	Swimming, Volleyball, Soccer, Basketball and Softball will be available during leisure time.
COMPUTER HARDWARE	WANG and RADIO SHACK TRS-80 computers with screens, cassettes, diskettes, printers, card reader.
LOCATION	The computer camp is located at the Grand View Lodge, Moodus, Connecticut 06469.
CAMP FEE	$295.00 for one week, or $590.00 for two weeks includes books, supplies, camp shirt, insurance, room, board, and for two week campers a trip to Riverside Amusement Park.
DATES	1st Session — July 19 to 24; 2nd Session — July 26 to 31. Two week session — July 19 to 31.
DIRECTOR	**Michael Zabinski, Ph.D., Professor,** Fairfield University, Fairfield, CT. Dr. Zabinski serves as a consultant to school systems throughout the country. He is associated with Radio Shack and has written several texts on computers. His most recent book is, "Introduction to TRS-80 Level II Basic and Computer Programming," Prentice-Hall, 1980. At camp he will be joined by computer experts who specialize in teaching computers in elementary and secondary schools.

"Our philosophy is to motivate students by presenting material in exciting, meaningful ways with examples they can relate to and identify with."

For further information contact National Computer Camp, Grand View Lodge
P.O. Box 22, Moodus, Connecticut 06469

(317) 742-7388

Micro Data Base Systems, inc.

P.O.BOX 248 LAFAYETTE, IN 47902

NEWS RELEASE

For Z-80, 6502 and 8080 based micro-computers, Micro Data Base Systems, Inc. has recently released a hierarchical (tree-structure) data base management system (HDBS). This new product is offered at $250 for the Z-80 version and $325 for the 6502 and 8080 versions, and contains many of the features available in the earlier released CODASYL network data base systems (MDBS). HDBS, written in machine language for maximal execution efficiency and minimal memory usage, contains commands to add, delete, update, search and traverse the data base. Users can define set relationships between record types in a number of different ways including sorting on various keys and FIFO, LIFO, NEXT and PRIOR orderings. Read/write password protection is provided at the File Level. Both MDBS and HDBS routines are callable from BASIC, FORTRAN, COBOL and machine language. Other host language interfacing is in progress.

The major differences between HDBS and Micro Data Base Systems' full network data base system (MDBS) are in the complexity of the data structure and associated data manipulation routines, certain data security features and the price (MDBS lists at $750). Given the difference in price, if the software applications only require a hierarchical structure, clearly HDBS would be the choice. Attempting to use HDBS where the natural structure is a full network requires that some data fields be repeated in several different records. This data redundancy introduces extra costs in loading and retrieving program and in storage. Whether the extra cost compensates for the savings obtained in purchasing HDBS must be evaluated by the user. MDBS is upward compatible with HDBS and such an upgrade costs $550. The HDBS-MDBS manual can be purchased alone for $35.

Micro Data Base Systems announces an important new utility for the use with its hierarchical data base management system (HDBS). This utility, called the Schema Redesign System (HDBS.SRS allows the user to modify important elements of an existing data base structure easily. The system permits the user to add new fields, expand the size of the data, base, and rename fields, records, and sets. The redesign is performed dynamically, relieving the user of the chores of offloading the data base, changing the structure, and then reloading the data base. The user may now modify major features of the data base faster and more easily.

A related product, the Dynamic Restructuring System (or MDBS. DRS), performs similar functions for a full, network data base system (MDBS). In addition, DRS allows the user to add new passwords, records types, and set types, to have new owners and members inserted into a set and to alter existing access links.

Also, passwords, fields, records, owner and member records, and whole sets can be deleted from the schema.

SRS and DRS are both available for most of the operating systems and languages with which HDBS and MDBS are interfaced. SRS is currently priced at $150 (US and Canada) and DRS is priced at $300 (US and Canada).

THE ANSWER MAN

Some answers (and maybe a question or two) to some letters which appeared in Proteus News, Vol. 3, #3.

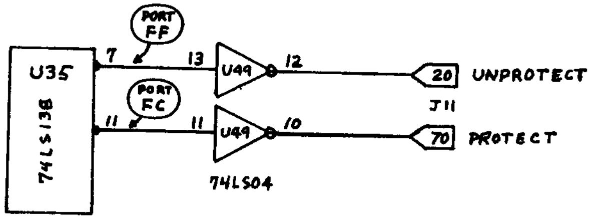
POWER BLINKS IN NEPAL

Dr. Henderson asks how to "unprotect" a PTC 8KRA memory board which he has jumpered to come up in the "protected" mode after a power failure. The difficulity is that the Sol does not use pins 20 and 70 of the S-100 buss which MITS (of Altair computer fame) originally intended for this purpose.

Answer: The trick for doing this for the 8KRA (and other boards too) can be found in Solos News (the former name of Proteus) Vol. 1, #5. (You did order your back issues didn't you?) In looking at Ron Parson's article on how to configure the Sol to jump from the Helios controller to the Tarbell controller, it can be seen that he connected an unused output port to feed a pulse to a latch which does the switching. The Sol has two output ports, FC and FF, which are available but not used. A look at the schematic for the Memory & Decoder curcuits (X-16) will show two empty connections on IC U35, a 74LS138. Pin 11 is port FC and pin 7 is port FF. An OUT <port> instruction will provide a negative going pulse to the corresponding pin. An inverter is necessary to change it to a positive pulse which is the required unprotect signal to be applied to pin 20 of the S-100 buss connector. See the diagram below.

Answer to question you just now asked: Yes, this same trick can be used to ring bells, sound buzzers, blow horns or even control the world from your Sol! Of course, some sort of interface may be required.

MEMORY PROTECT/UNPROTECT CIRCUIT



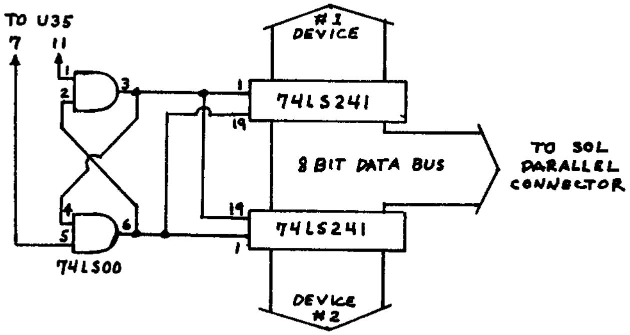
NOTES: U49 HAS 2 SPARE INVERTERS WHICH MAY BE USED.
ON REV. E SOLS, J11 PIN 70 MAY BE GROUNDED.
THIS GROUND MUST BE CUT.

ONE PORT AND TWO PRINTERS

John Gorman wants to hook up two printers (or any two devices) to the same port and not have them interfere with one another. Now isn't it amazing! The solution to Dr. Henderson's problem above is also the answer for John's. Who would have thought the Sol was so versatile.

Answer: John will need the full latch circuit with a few buffer IC's to handle the data and direct it to the proper device. An OUT <port> instruction to, say port FC, will direct the data one way by flipping the latch and enabling one of the buffers, while, an OUT FF will flip it the other way, disabling the first buffer and enabling the second. This can make for snazzy performance since both devices can be "on line" at the same time. Imagine a printer putting out some text uhile a plotter simultaneously draws a graph of the same data! The driver, of course, would have to contain the proper OUT instructions to do the job but that's a software problem and that's easy, right?

CONTROLLING MULTIPLE DEVICES FROM PARALLEL PORT



NOTES: AN "OUT FF" WILL TURN ON DEVICE #1 WHILE AN "OUT FC" WILL FLIP TO DEVICE #2.
THE HANDSHAKING SIGNALS ARE NOT SHOWN AND MAY REQUIRE AN ADDITIONAL IC.

But what if John doesn't need to operate both printers at the same time? In that case the solution can be very simple. I have a Spinwriter and a TI 810 connected to my Sol's serial port. I found that I can just wire their cables together into a common plug which goes to the Sol. As long as I don't turn them both on at the same time it works fine. If I did, their "handshaking" signals (the voltages which tell the Sol it's OK to send data or not) might fight each other and destroy an IC or two. I can even use seperate drivers for the two printers since the Sol only "sees" one of them at a time.

QUME PROBLEM

Jim Michaels problem is that, when he switches from printing with WordWizard to printing with DBASIC, he has to open up the Sol (and his Qume printer) and change the baud rate switches. WHOA!! Something's wrong here! Both WW and BASIC use the same serial port and the same UART for output so why have to change anything? Ah, I think Jim forgot to mention something very important. I think he forgot to tell us that when he tries to print a BASIC listing, for example, the output "overruns" the printer. The first few lines print out OK but then things seem to get all messed up. By trial and error Jim found out that when he set the baud rate down to a lower value the printer did everything correctly. Right Jim?

Answer: The solution to Jim's problem is handshaking. When Jim uses his WordWizard word processor for printing he uses a "print driver" which comes as part of the WW package. This driver is a "high level" driver which means it contains lots of bells and whistles. One of these whistles is a routine which detects when the printer has received enough characters and just can't take any more. The printer signals the Sol of this fact by changing the voltage level on one of the wires connected to the Sol's serial port. The Sol has to be able to understand this signal in order to stop sending characters until the printer catches up. It does this by testing the "status word" of the serial port. This test is performed by software contained in the driver. (When the printer can accept another batch it changes the voltage level on that wire back to its former value.) Now, when Jim tries to print from disk BASIC, he uses the PTDOS command "OUT P" which means - send all output to the Sol's serial port. The output now goes to a different driver, a bare bones one, which contains no test for the status word. The serial port driver contained in the Solos ROM is also a bare bones type which tests only that the Sol does not send characters out faster than the selected baud rate. It performs no tests whatever of the printer. As a matter of fact, you could unplug the printer during the middle of a printout and the Sol would merrily continue sending characters as if nothing had happened! Jim's fix of lowering the baud rate only means that he forced the Sol to slow down sending characters to a rate at which the printer's buffer would never become clogged. This is very inefficient and we can do much better. Jim also wants the output echoed to the video and we can fix that too.

What Jim needs is a "custom driver." Custom means it is usually written by the user rather than the manufacturer. Jim is fortunate however. His driver is already written for him and is contained in a file on his WW system disk called PRINTER. He can copy it over to his BASIC disk by use of the PTDOS "GET" command. Put the WW system disk in drive 1 and the BASIC disk in drive 0. Then type after the PTDOS prompt (*): GET I=/1,/0,PRINTER

Jim can now enjoy trouble free output from BASIC by putting a statement ahead of the part he wants printed such as:

```
          80 SET OF=PRINTER
```

When it is desired to send the output back to the screen, insert a statement such as:

```
          220 SET OF=#1   (#1 is the screen driver in PTDOS)
```

But Jim wants to see an echo of what is going to the printer, on his video screen. OK Jim, but now you are entering the realm of "custom machine language programming." But hark! It's already been done for you. If you look at Vol. 2, #3, p. 18, of Proteus News at the OUTPUT routine given in the listing for the Northstar I/O, you can see how it's done. There is another example in the same issue on page 20. A third method is given below:

```
C900 3A 19 C9   OUTPUT LDA FLAG    Get the print flag
C903 FE FF             CPI 0FFH    Test print flag
C905 CC 0C C9          CZ  PRINT   If match, go print
C908 AF                XRA A       Set pseudo port for VDM
C909 C3 1C C0          JMP 0C01CH  Solos output routine
                *
C90C DB F8      PRINT  IN  0F8H    Get status word
C90E E6 A0             ANI 0A0H    Mask off all but UART and CTS bits
C910 FE A0             CPI 0A0H    Test for both bits present
C912 C2 0C C9          JNZ PRINT   If missing, keep looping
C915 78                MOV A,B     Put character in A
C916 D3 F9             OUT 0F9H    Send it out
C918 C9                RET         Return
                *
C919 00         FLAG   DB  0       Store initial 0 for no print
```

Notes: 1.) In order to have selective printing, a flag byte is stored at location C919H (51481 decimal). ENtering an FF at that location will send output to the serial port. Any other value turns off printing. Output will go to the VDM in any case.

2.) The test for A0 tests both the Transmit Buffer Empty bit (80H) and the Clear To Send bit (20H). The CTS signal is the one from the printer which causes the Sol to halt until the printer is ready. At the Sol end of the cable it is called CTS but, at the printer end, it is called DTR (Data Terminal Ready). Often, the "Reverse Channel" pin is used at the printer end rather than DTR. REV CHA inverts the logic of the signal if that is what is desired. It should be connected to pin 5 on the Sol's serial connector.

3.) The line from the printer should be "active low". This means that when the printer cannot accept characters, it brings the CTS line to some negative (usually -12) voltage. When the printer is ready, it should bring the CTS line to some positive voltage. (usually around +12) There is a reason for this. The Sol has internal circuitry which "pulls up" the status bits to a positive value when nothing is connected to the serial port. Using the "active low" convention on the CTS line means that this print routine will never "hang" if the printer is turned off or is not connected. Some programs will crash if you try to RESET them out of an infinite status loop. On my Spinwriter, I set contact 5 of SW 1 to "ON" (up) on the control panel circuit board (G9BNF). I then connect the Reverse Channel line, pin 19, to the Sol's pin 5. On my TI 810, I connect the Reverse Channel, pin 11, to the Sol's pin 5.

4.) If this program will be used with PTDOS it must be preceded by a driver table. See Section 9 of the PTDOS manual. An example of a minimum size device driver was given in Solus News Vol. 1, #4, p. 15. If it will be used with Solos, then the procedure given in the Solos manual should be followed. If the program is loaded at address C900, as given in the listing above, then the Solos command to set the custom output routine should be entered as: SET COUT=C900 followed by: SET O=3. Remember that any time you RESET the computer these parameters will be wiped out and must be reentered.

ELECTRIC PENCIL PROBLEM

Roger Doran can't get his Centronics 779 to space properly when using his Electric Pencil word processor.

Answer: The EP puts out line feeds to accomplish vertical spacing. Many printers do not recognize LFs but only carraige returns. I seem to recall that Michael Shrayer published some notes on where to replace all those LFs (0AH) in the program with CRs (0DH). Failing to find that info, I suggest that you search the program code for 0A's and selectively change them to 0D's one at a time. Try the program on a test file after each change until you find the right ones. I think there were about half a dozen locations which required changing. A suitable search program can be found in Proteus News Vol. 1, #6, p. 7. and also Vol. 2, #4, p. 15.

WORDWIZARD PRINTER DRIVERS

Earl Dunham asks about the availability of Diablo, Centronix, Spinwriter and Selectric device drivers for use with WordWizard.

Answer: Drivers for all of these printers are available on either the WW update disk, (available through Proteus) or in the Helios Software Library. On Helios library disk #3 were drivers for the TI 810 and the Spinwriter. The source code for both drivers was given and the one for the Spinwriter included the space-averaging routines. (there will be an updated version of this one on a future Helios Library disk). A general purpose driver for the Diablo appeared recently in Vol. 3, #1, p. 19, of Proteus News.

COMMENTS

Many thanks to Bill Blomgren for his clarification of the PTDOS "READ" command. Now, if someone would just explain the difference between "IMAGE" and "WRITE".

I would suggest that contributors of program listings to Proteus News should please submit <u>assembled</u> listings of machine language programs - formatted and commented if at all possible. The object code really helps if the mnemonics become hard to make out after reduction and printing in the newsletter. I considered particularly good examples of format and clarity in the Vol. 3, #3, newsletter to be Melvin Dalton's listing on page 5, and Daniel Hunt's on page 13.

As can be seen by replies to some of the members problems above, it really helps to give as much information as possible when writing for help. I wanted to help Dr. Isard with his instrument interface problem but there just wasn't enough information to even get started. I have noticed also, that many of the answers to problems can be found in previous issues of Proteus.

By the time you read this, Northstar should have its new, single board computer out. It's a attempt to cash in on the TRS-80 market. For us Sol/Northstar users, there is a new DOS, BASIC and Utilities. It is called version 5.2(S) or (DQ) and is COMPLETELY RELOCATABLE! Alas, some of the new Utilities (RAMTEST) are written in Z80 code. I think the handwriting is on the wall and Northstar is going to abandon us 8080 types.

QUESTION

I was told by a former technician of PTC that the Helios could not be used with a Z80 system because of timing problems. (at 4 mHz) I know from comments I have read in the newsletter that some of you out there are doing the impossible! Any information about how the marriage was accomplished would be appreciated. Thanks.

Contributed by: Joe Maguire
P.O. Box 3742 DT
Anchorage, AK 99510

A Report on Northstar Computers' Hard Disk System

by Joe Maguire
P.O. Box 3742 DT
Anchorage AK, 9951Ø

This report on Northstar's new HD-18 hard disk storage system consists of some good news and some bad news. First the good news: My overall opinion of the system is good with excellent software support. Now the bad news: The system will not work with the Sol computer.

Northstar lists the following requirements for the HD-18.

1. A Z8Ø CPU. (the HDOS is written in Z8Ø code)
2. Processor speed of 4 mHz. (required for data transfer rate)
3. 48K of contiguous memory with 56K recommended.
4. Parallel port interface.

As can be seen from requirements 1 and 2, this rules out the Sol or any 8Ø8Ø CPU based system. The HDOS (hard disk operating system) occupies the first 18K of memory beginning from address 1ØØ hex. The HDOS contains all the necessary code to permit data transfer operations from memory to the hard disk or to the minidisks or to any combination of the three. Also included in the HDOS are all the Monitor functions which are in a separate file in the non-hard disk system plus a large number of new commands such as EB (examine byte), EM (examine memory), EP (examine port) and so on. It is obvious why Z8Ø code was used to write the HDOS, it just wouldn't fit in 18K otherwise. (Application programs such as Basic can overlay part of the HDOS increasing the remaining available memory.)

I got the distinct impression from the HD manual that NS was trying to push the use of their Horizon computer with the system but I also feel that the software could be rewritten to accommodate the many 8Ø8Ø based users of Northstar products. Perhaps after the initial rush of orders is over NS will get around to providing a version of the HDOS for them.

In addition to my Sol, I have an Horizon so I was able to make use of the system. The drive is a Century Data Marksman which is contained in a sealed unit. The capacity of 18 megabytes allowed me to tranfer my 5Ø minidisks of programs and data over to the HD with still some more room to go. This new found convenience is not without its hazards however. A disk crash can wipe out years of work! NS has considered this possibility and has included a well thought out series of backup and recovery programs. It takes only a few minutes a day to accomplish using the minidisks which are expected to be available with the system. This is in marked contrast to the 3Ø to 4Ø minute tape backup I observed with another system.

I will have a more detailed report of the system published in the new Northstar Users Association newsletter for those who can make use of it with their existing hardware.

MOVING SOLOS TO FØØØ

Relocating SOLOS to F000 (PROTEUS News, V.2,#3,p.2) seems like an ideal way to expand into a CP/M system larger than the 48K allowed with SOLOS at C000. Bob Goodman's hardware instructions were easy enough to follow (jumper pins 9 & 12 of U22, pull pin 1 of U24 to +5v through a 1.5K resistor), but I did have some difficulty relocating SOLOS. The source code from the CP/M User's Group had been modified slightly from the published listing of SOLOS. Since some programs published in PROTEUS call obscure routines in SOLOS, I returned the source code back to its original form to maintain adress compatability, except for the high order byte. Changes made to relocate SOLOS included changing the origin from C000 to F000, changing the VDM address space to FC00, and changing the start of system RAM to F800.

A less obvious change that was necessary was in the 'clear screen' routine, ERAS1 (C0DA-C0E4). SOLOS assumes that the VDM is at CC00-CFFF. It checks for the end of the screen by comparing the high order byte of the current cursor position to D0 (the first address after CFFF). If you move the VDM to FC00, this routine fails; in fact, the 'clear screen' function doesn't work even if you change 'D0' to '00'. It fails on the instuction JC (jump on carry) because you will never get a carry by subtracting '00' from any address. The simplest patch was to change the JC to a JNZ (jump on non-zero).

For the adventurous: I have sent a) acopy of the relocated SOLOS source code (under CP/M) and b) the object code of same (on CUTS tape) to our program librarian, Lewis Mosley.

For the lazy but patient: A rumor from a usually reliable source (Lee Felsenstein) suggests that an un-named group of co-conspirators (including Lee Felsenstein as one of the henchmen) are involved in a 'Save Our SOLs' project. Said project is a rewritten, expanded SOLOS relocated to F000 on a 'smart' Personality Module, i.e., the modification will allow you to switch between SOLOS at C000 and at F000 simply by switching PMs, no external switching required. Watch PROTEUS for further details, or contact Lee at Golemics, 2608 8th Street, Berkeley, CA. 94710.

Now - HELP! How do I go about changing all my ProcTech software so that it runs with the VDM at FC00?

Bob Stek, 19 Mayfield Road, Regina, Saskatchewan Canada

ANOTHER NOTE ON PSCAN. by Grayson Evans

Previous articles that have delt with the PSCAN routine in PTDOS have shown the various methods of accessing input command lines. You may decide to do your own PSCAN routines by accessing the input buffer directly. In PTDOS 1.5 the input command line is stored in a buffer at 99F0H. The input line ends in a CR.

If you need to alter the buffer be sure and copy it to a work area in your program. Be sure to use the RESOP system call to return from your program to PTDOS since this will "flush" the contents of the buffer before proceeding.

WHAT DISK SHOULD I GET?

That's a question that is asked often in the letters to Proteus. Usually the writer wants a disk system that will plug right into the Sol, work reliably, and not take ANY modification whatsoever to install the hardware or software. I usually recommend the Thinker Toys disk, ordered with CP/M specific for Sol. I've also been told that Vista makes a disk with CP/M that will also meet the qualifications of a plug-in-and-go disk. If anyone has another comment in this regard, please write a letter for publication.

Stan

EQUIPMENT REVIEW

Memory boards and Z80 processors.

I have some experience with a few memory boards that I have not seen reviewed in PROTEUS. Since I am running PTDOS with the Helios, memory boards have been a pain in the whatever. After trying at least eight 16K and 32K boards in my system trying to find a few good boards I have a big list of boards that dont work and a list of boards that do. Some you have already read about.

Calf. Computer Systems XVI 16K static board work fine. It is loaded with switches for selecting this and that bank select options, addressing 4K blocks, etc. It has LED's to indicate board select and board active. Manual is very good.

The old Altair 88-16K memory board also works like a champ. This board has been around for a long time and can easily be found in the classified sections at reasonable prices. This is a 16K static board and can be addressed at 16K increments.

I have not had good luck with Godbout Econorams, Cromemco's 16K bank select boards, or E.C.T 8K static boards.

For those of you who would like to run a Z80 processor board in something other than a Sol using the Helios drive I would recommend Cromemco's Z80 CPU board (NOT the single board computer which don't work). It is the ONLY Z80 CPU board I have ever tried that works perfectly with the Helios. It goes to a great deal of trouble to exactly duplicate the Altair S100 signals including interrupt enable. The S.D. Sales and Ithica Audio CPU boards will not work.

Grayson Evans.
1243 Manchester Ave.
Norfolk, Virginia 23508

MINOR INPROVEMENTS TO THE HELIOS DISC CONTROLLER BOARDS
by Grayson Evans

This article describes a few minor improvements that can be made to the Helios disc controller boards that will increase their reliability and life span.

1. Curing heat problems.
As you know (if you own a Helios disk system) the disc boards run quite hot due to overworked regulators and a high density of IC's. The heat disipation capabilities can be improved by making the following modifications:

On the Controller board, install extra heat sink area to the 5V regulators by gluing (with epoxy) Thermalloy THM6010 heat sinks to the top (plastic) of the regulators. Apply a thin even coat and press the heat sinks across the regulators until the epoxy is dry. They will fit nicely within the present heat sink. Other heat sinks could work just as well as long as the total height is less than 3/4".

Also, since the Fairchild 9403 FIFO chips run very hot apply a Thermalloy THM6085B to the top of these chips in a similar manner. These extrusions are designed to be bonded to the tops of chips and although these are made for 24 pin chips, they work fine for these odd sized IC's.

On the Formatter board add a THM6106PB heat sink to the present naked 5V regulator. Even though this regulator does not get very hot the heat sink will add years to its life span.

All of these heat sinks are available from TRI-TEK Inc. in Arizona.

2. Replace existing 50 pin headers.
If you have had to remove the boards several times from your computer you know what a fun time it is to unhook the 50 pin ribbon connectors from the boards. After having broken pieces from all the connectors by using a screw driver to pry the connectors out I decided to fix this problem once and for all. Removing the connectors may be made a joy by installing new 3M right angle headers with built in ejectors. This allows the connectors to effortlessly removed eliminating stress from the cable, connector and header. The 3M connectors are part number 3433-1302. These allow the use of strain relief connectors if you prefer. They cost about $8.00 each.

To install the new headers it is necessary to remove the old ones. This must be done using a vacuum powered solder sucker. Do not attempt to remove these connectors using solder wick or the various spring loaded solder suckers. Find someone who has a Weller, Pace, or similar powered vacuum unit designed for this kind of work.

3. Upgrade the bypass caps.
While you are having the header removed also have the .04uf disc ceramic bypass capacitors removed. Install in thier place smaller size and larger value caps. I recommend caps such as Mallory C12C104M5U1CA at about $0.19 for 50 or Sprague 2CZ5U104X0050C4 at about $.24 for 50. These capacitors are small (don't stick up above the chips), .1uf 50v ceramic caps, designed for bypass. The bypass capacitance on the Controller board is a little scarce due to the limited real estate. Using .1uf caps instead of the .04's supplied will more than double bypass capability.

While these mods will not move data one nano-second faster they should extend the useful life of the boards considerably.

MODIFICATION FOR USING THE CUTS BOARD AS A SERIAL PORT

By Steve Eigsti

With very little work the CUTS board can be used as a 300 baud TTL serial port suitable for driving a modem. The modification defaults to 1200 baud cassette operation, and is selected by software by setting the tape speed to 300 baud. Since the 300 baud data will be accepted from the outside world the low speed cassette input operation is disabled. It was not a problem for me but if you have cassettes recorded at the low speed you'll need to convert them to 1200 baud. You can, however, still record with this modification at 300 baud.

I will assume you know what a modem is and how it works, and will only describe the nuts-and-bolts procedures to modify the CUTS board. Refer to the CUTS schematic in the manual and my Figures 1 and 2 for this explanation.

There are two spare tri-state gates on U-17 which will be used. One will be connected in the line from U-22 pin 6 to UART pin 20 (serial data in). See Figure 1. The other is connected in the line from U-22 pin 10 to UART pin 17 (read clock in). The enable for these gates is tied to pin 2 of U-13. During 1200 baud operation (cassette) this pin is low, enabling the cassette circuit (U-4) to pass data to the UART, and the recovered read clock data goes to the UART receive clock.

For 300 baud operation several simple external circuits buffer the data (Figure 2). When 300 baud is selected U-13 pin 2 goes high and pin 3 goes low. Both tri-state gates are disabled. The UART can now accept external TTL data without interference from the cassette circuit (U-4). The write clock on UART pin 12 is connected to the receive clock through an external tristate gate (Figure 2). This gate is enabled by the low of U-13 pin 3.

There are five external signals connected to the CUTS which I placed on the S-100 bus on pins unassigned (at the time). Check your own system to make sure these are not used;

1. Serial data in (TTL), S-100 pin 17. This is the 300 baud from the modem.

2. Serial out data (TTL), S-100 pin 15. From the UART to the modem.

3. 300 baud enable, S-100 pin 16. Low=300, high=1200. Used to enable the external gates as well as those on the CUTS board.

4. Write clock out, S-100 pin 56. This is 16 times the baud rate and generated by the CUTS board.

5. Read clock in, S-100 pin 56. The write clock is fed back in to drive the read clock on 300 baud, since we disabled the read clock.

Remember, this is TTL not RS-232. It's up to you to provide the modem and coupler. Check the back issues of Kilobaud and Byte, among others, for details on modems. There is a low-cost modem board available from Electronic Systems in San Jose which recently received good reviews in Kilobaud.

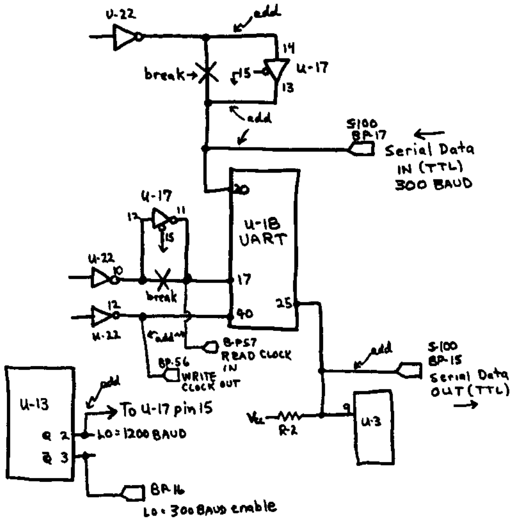


FIGURE 1. MODIFICATIONS TO CUTS BOARD FOR 300 BAUD SERIAL DATA

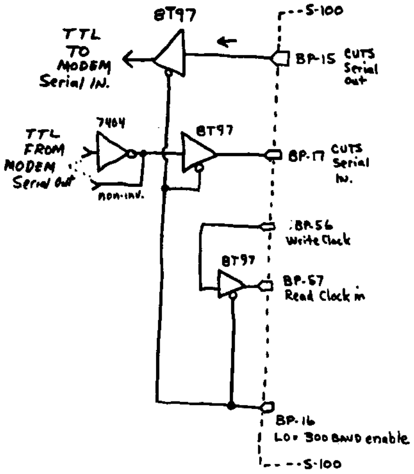


FIGURE 2. EXTERNAL GATES FOR 300 BAUD CUTS MOD.

Two more mods...

There are two more mods you may wish to make. First, I recommend changing the CUTS clock to the S-100 clock on pin 49 rather than Phase 2. That way, if you upgrade to 4 MHZ the clock will always have 2 MHZ.

Second, if you use SOLOS there is a terminal driver built into the software but it wants F8H as the serial port, not FAH as in CUTS. The status latch U-13 can also set the port address if you connect pin 2 of U-13 (low=1200, high=300) to pin 1 of U-14. This has the effect of setting the address to F8H when the low speed is selected. If you do not have SOLOS but CUTER instead you will have to write a terminal driver anyway and can specify FA. You may have to write one anyway if you need to send parity.

With a homebrew modem (salvaged from a Mits board) I have used this CUTS driver on three computers, including Micronet, without any problems. In fact, I even used it for ham RTTY and interfaced it to a Dovetron modem and converted Baudot to ASCII. The CUTS board works nicely even with 5 bit characters in Baudot.

INTERFACING A SOL COMPUTER TO AN H-14 PRINTER

By Bill Supynuk

The Processor Tech SOL computer is an excellently designed computer and the Heathkit H-14 printer with its features is good value for its price. However, interfacing the two can present a problem until one takes into consideration that the SOL is configured, at its serial port, as a terminal and not as a computer. The solution is to reverse the usual RS-232 pin configuration (see Figure 1).

	H-14		SOL	
PGND	1	....	1	PGND
SOUT	3	....	3	SIN
SIN	5	....	2	SOUT
RTS	7	....	5	CTS
CTS	9	....	4	RTS
DSR	11	....	20	DTR
SGND	13	....	7	SGND
DTR	14	....	6	DSR

Figure 1: Wiring Configuration

The easiest way to change the pin configuration is at the H-14 plug of the interfacing, or connecting, cable. Position the H-14 plug on a flat surface with pin #1 at the top and the shiny parts of the pins peeking up at you through their rectangular holes. Using a pointed tool depress, on the left side of shiny part, the tiny little rect-angulular tabs holding the female pins in the square holes and simultaneously tugging at the wire, with your other hand, pull the pin out of the plug. Interchange the appropriate pins, two at a time. The interfacing chart on page 8 of the H-14 Operations Manual will identify the pins by number and wire color.

With the baud rate switches on both the SOL and the H-14 set to 110 and the SET O=1 command (use only at 110 baud) and a DUMP to memory command, you can test the interface to see if it is working. (Make sure the SOL's serial word definition switches are set to NO PARITY, 2 STOP BITS, and 8-BIT DATA WORD LENGTH).

```
            SET O=1      <CR>
             DU 00 FFFF <CR>
```

The H-14 should print out a beautiful set of characters. If it doesn't look for a wiring error.

If a baud rate higher then 110 is to be used, then hand-shaking must be employed. Also, the characters being printed by the H-14 will not be displayed on the monitor unless a spec-ial routine is written and included in the handshake program (see Figure 2).

```
CB00 DB F8       0001  HANDS   IN      PTRST      ; get staus byte
CB02 E6 20       0002          ANI     MASK       ; get status bit
                                                  ; (1=free)
CB04 CA 00 CB    0003          JZ      HANDS      ; go back if busy
CB07 3E 01       0004          MVI     A, 1       ; select serial
                                                  ; port  (A=1)
CB09 CD IC C0    0005          CALL    AOUT       ; output character
CB0C AF          0006          XRA     A          ; select video
                                                  ; screen (A=0)
CB0D C3 1C C0    0007          JMP     AOUT       ; output character
CB10             0008 *                           ; NOTE: uses AOUT'
                                                  ; S RETurn
CB10             0009  PTRST   EQU     0F8H       ; SOLOS status
                                                  ; port
CB10             0010  MASK    EQU     020H       ; bit 6 is the
                                                  ; status port
CB10             0011 AOUT     EQU     0C01CH     ; SOLOS OUTPUT
                                                  ; routine
AOUT        C01C       0005  0007
HANDS       CB00       0003
MASK        0020       0002
PTRST       00F8       0001
```

Figure 2: Handshake and Display Program

To enter the Handshake and Display Program go into SOLOS and type in the following:

```
CB00 DB F8 E6 20 CA 00 CB 3E 01 CD 1C C0 AF C3 1C C0/ <CR>
```

Then SAVE the program on cassette tape calling the program "PRINT", or similar:

```
SAVE PRINT      <CR>
```

You now have a permanent copy. To utilize the program use the following routine in SOLOS:

```
GET PRINT       <CR>
SET COUT=CB00   <CR>

XEQ BASIC  (or, PILOT, etc.)
```

In BASIC (Proc Tech Extended) use the SET OP=3 command to start the H-14 printing and SET OP=0 to stop.

In SOLOS use O=3 and O=0 as the print commands.

A note about the baud rate: both the SOL and the Heathkit H-14 will run very efficiently at 2400 baud. So be sure you set the baud rate switches on both machines to that rate.

On the SOL also set the serial word definition switches to NO PARITY, 1 STOP BIT, and 8-BIT DATA WORD LENGTH.

One of the features of the H-14 is the defining or chang-ing of the width of characters (and thereby the size) and the the numbers of lines of characters per inch (also changing the their size). This can be done in BASIC by using the PRINT command encorporating the ESCAPE key character and a specified letter within quotation marks, such as

```
        100  PRINT "&(x"
   or   100  PRINT "&(y"
```

The "Top of Form" command, to instruct the H=14 to roll the paper up to the top of the next page, can also be done through BASIC by

```
100 PRINT CHR(12)
```

The character control codes are in the H=14 Operations manual for your reference.

Bill Supynuk,
8014-163 St., Edmonton
Alta, Canada T5R 2N3

Happy printing!!

DEVICE DRIVERS EXPLAINED

Grayson Evans
1243 Manchester Ave.
Norfolk, Virginia 23508

This article will try to provide a clear and detailed explanation of the use and operation of device drivers. It gives on overview of thier purpose, of thier design and a detailed review of the use of entry points and two examples of simple device drivers - one that does input and one that does output.

The concept of device drivers is not new. Anyone familiar with Digital Equipment Corp. operating systems and/or Unix (which was designed to run on DEC computers) will be familiar with thier use. I dont know how far back the origin of device drivers goes or who used them first but I suspect they evolved slowly as the need to deal with ever more complex I/O devices in a simple and consistant way became a pressing problem.

Device drivers attempt to solve some of the I/O interfacing problem by providing software which allows I/O devices (thru I/O ports) to "look" to system and user written software like, and enjoy the same privlages as, the mass storage file system. In the case of PTDOS this is the floppy disc system. Thus functions which the operating system allows to be performed on the file system may potentially be performed on any interfaced I/O device. This allows a common set of system protocols to be used for access to both disc files and I/O devices.

The device driver capability provided in PTDOS is a good one but unfortunately a good explanation of how to take advantage of the capability and how it works is not provided. After a brief introduction and entry point explanation in section 3 of the PTDOS manual, the user is left to fend for himself with the eye-straining CUTS example found in Appendix C. Information about device driver operation can be found in Section 3.8, scattered throughout system calls in section 2, Appendix A, and Appendix C.

The information is this article has been gathered from the PTDOS manual, trial and error, experimentation, calls to PTC, and debuging several device drivers.

OPERATION AND DESIGN

A device driver is an executable program which has a name and resides as a file on the disc. The program is written to interface between PTDOS and an I/O device attached to the system Since PTDOS cannot be written to anticipate all the gadgets attachable to its host nor can its code be modified to handle the I/O requirements of these gadgets, the device driver provides a standard set of entry points for PTDOS and handles the transfer of data and control to and/or from the necessary I/O ports to control the gadget.
Since the device driver program resides on the disc under some user supplied name and is thus known as a device file.
The device file looks like any other image file on the disc with the exception that it is given a TYPE of "D" by the user.
In order to transfer data to or from the I/O device the user or system program performs the operation on the device file.
Since the device is now treated as a disc file, the file must first be opened. This is where PTDOS discovers the file is a device driver and treats the loading of the file differently (the description of the OPEN system call on page 2-7 gives a detailed explanation). The fact that the file is a device driver is never known by the program requesting access to the file. PTDOS is responsible for for knowing whether to transfer the data to and from a track on the disc or to and from the now loaded device driver.

The device driver program is written as a group of subroutines. There are as many subroutines as there are operations to be performed by the driver. The first 25 bytes of the program are known as the DRIVER TABLE and consist of 11 possible entry points, or addresses, of the subroutines in the program. There is also a block size and ITO value in the table that will be discussed later.
The driver table is the same for all device drivers (i.e. all 25 bytes must be there) and contains entry points for all possible operations that may be performed on a "file" whether its a front panel light or a random access disc file.
The table is compiled as a set of addresses. The addresses are used by PTDOS to know where to find the applicable subroutine. If a particular operation is not supported by the device driver, the entry must contain zeros. It is not necessary for a device driver to perform all 11 operations! A device driver for the front panel lights cannot possibly perform a seek. The two example programs illustrate various combinations of supported operations.

DRIVER TABLE

Page 3-16 of the PTDOS manual gives an explanation of the driver table organization. Notice from the table on the page that the subroutines in the driver have either one or two possible returns back to PTDOS. These are known as the CALL+1 return and the CALL+2 return.
The CALL+1 return refers to a normal RET instruction return to the first byte after the last CALL instruction. A CALL+2 return refers to a return to the forth byte after the last CALL instruction "performed by first triply incrementing the address on the top-of-stack and then performing a RET...".
If an operation has two possible returns then by PTDOS convention the first return is taken if some error or abnormal condition occures (an EOF is treated as an abnormal condition). The second return is taken if the operation was performed successfully.
Note that the REWIND, END-FILE, CLOSE, and INITIALIZATION operations only use one (CALL+1) return.

A typical access sequence to read a block of data from a device might be as follows:
An executing program, after opening the file, would make a system call to read a block of data (RBLOP). Using the file number supplied in the call, PTDOS knows the read is directed to a loaded device driver. PTDOS obtains the address of the code to perform the operation from the READ BLOCK entry (DTRB) in the driver table and transfer control to that address with a CALL instruction. The device driver subroutine then performs the read operation from the physical device. If the read was successful the subroutine returns to PTDOS thru the CALL+2 return. PTDOS then returns control to the calling program passing information about the operation in various registers. If PTDOS found a value of zero for the address of the read operation in the driver table, it would indicate the device driver was not capable of performing a read operation and PTDOS would issue a "ILLEGAL DRIVER ACCESS" message to the console device.

The two remaining items in the driver table are the BLOCK SIZE and a thing called the ITO. These are not addresses but are

G. EVANS CONTINUED FROM PAGE 12

data used by PTDOS in managing the driver operation. The 16 bit value provided in the BLOCK SIZE entry is the "block size" of the device and is used as a buffer for data read and/or written by the device driver. When the driver is first opened for use PTDOS assigns a buffer in its SYSTEM MANAGED BUFFER AREA equal in size to the value in BLOCK SIZE. When a read or write operation is performed by the driver the address of this buffer is supplied to the device driver subroutine in the HL register pair (see ENTRY value for each operation). Theoretically, the value given in the block size entry should have something to do with the block size of the I/O device (if the device is block structured). For example, if the device driver was interfacing a magnetic tape unit which wrote and read data in 256 byte blocks, the block size entry should be a minimum of 256. If the device is non block structured such as a printer or front panel lights, then almost any value greater than zero will work. A value of one if perfectly OK for a printer. If the interfacing device will accept or send data rapidly (such as another computer) then a large block size (1024 perhaps) would be more efficient.

The ITO entry stands for Immediate Transfer Option and is a one byte switch used by PTDOS. It has meaning only for write operations to the driver. If the entry is 0 (or off), then PTDOS waits until the driver buffer is full before a write entry to the driver is taken. If the ITO entry is 1 (or on) then a write entry is taken to the driver each time an executing program makes a write system call to the device file.

For example:
Lets assume a device driver that writes to a printer with a block size of 64 and a user program that writes 16 bytes at a time. If the ITO=0 then the user program would have to make 4 system calls to PTDOS to fill the device drivers buffer before PTDOS would call the device driver to print the buffer. If the ITO =1 then each write system call to PTDOS from the user program would cause PTDOS to call the device driver to print the 16 bytes.

Which option to use? Drivers that support block structured devices (mag tape, another disc, etc) should write a complete block at a time and should set ITO=0. Non block structured devices can use either option but if it is important to transfer the data as soon as its generated by a write operation (to a terminal, another computer, etc.) then set ITO=1.

Pages 3-17 thru 3-19 in the PTDOS manual give an explanation of each entry in the driver table. I would like to add a few words about several of the entries that may prove helpful.

*The phrase "Used by the ------- operation", where the dashes are the name of the operation, means "Used by the -------- system call described in section 2."

*Of the three read block entries, I have only been able to generate an entry to the DTRNB operation. I have tried various system software and system calls.

*The writing of a full buffer when ITO=0 is always taken thru the DTWBR entry. When ITO=1, the DTWB entry is taken.

*When an executing program issues a close system call to PTDOS, PTDOS will automatically empty any data left in the device drivers buffer to be written by calling the DTWB routine. PTDOS will then call the DTCLO routine.

*A more detailed description of the use of the DTCTL entry is found on Page 2-26 and 2-26 under the control/status system call.

*The DTINI entry is automatically called by PTDOS when the device driver file is opened.

EXAMPLES

Two examples are provided to illustrate device driver usage for data reading and writing. Unlike the CUTS example in Appendix C of the PTDOS manual, these examples have been kept simple and short for easy reading. They do work but would probably be elaborated to provide more useful device drivers.

The first example is used to interface a serial printer device. Notice from the driver table that the first three entries are zero indicating that the device does not read. The two write entries are covered by the same routine. According to the PTDOS manual the DTWRB entry is provided to do a read after the write and return the number of bytes read in the HL register pair while the DTWB entry only writes. These entry points are redundent since HL may simply be set to zero to indicate no read was performed.
The REWIND, SEEK, and CONTROL/STATUS operations are not supported. The EOF operation is covered by the driver for cases where the an executing program may ask for it. The The CLOSE operation is used by the driver to signal the end of printing and eject a bit of paper from the printer. The INITialize operation sets up the serial interface ports.
Of the operations provided by the driver only two are required: CLOSE and INITialization. The DTINI entry point must have an address since this entry is taken automatically by PTDOS when the file is opened. The DTCLO entry must be provided so PTDOS can close the file otherwise it will remain open until PTDOS is rebooted. Although these two entries must exist, they need not point to anything other than a RET instruction.
This driver has a block size of one and an ITO=1. This means that each byte writen by a program using the driver is printed immediatlely. Actually, with a block size of one, the value of ITO doesn't matter. The results will be the same.
Notice the EXIT ROUTINES. This code provides the CALL+1 and CALL+2 returns. Everything else in the program is dependent on the characteristics of the device.

There are a few points of caution when using drivers that write with ITO=0 and a block size is given that is greater than one.
* If the drivers buffer contains data but is not full and an executing program executes and END-FILE system call to the file, PTDOS will call the drivers write routine to write whats in the buffer but will not zero the buffer (or reset the buffer pointer the start of the buffer).
* A subsequent CLOSE system call by the executing program will cause PTDOS to write the contents of the buffer (by the drivers DTWB entry) and then take the DTCLO entry to the driver thus causing the buffer to be written twice.

The second program is an example of a device driver that reads data. In this case the program reads data form a paper tape reader of the "armstrong" variety attached to a parallel port. The driver reads the tape with no data conversion or intervention. An EOF condition is generated by setting one of the front panel data switches on.
This driver is designed to support the DTRB and DTRNB entry points. An entry for DTCLO and DTEOF is also provided to satisfy possible attempts at these operations for an executing program.
A block size of 32 has been arbitrarily chosen to buffer incoming data. The ITO option is not used for read operations.

There are a few points of caution when using drivers that read.
* When a read entry is taken by PTDOS to the driver, DE contain

G. EVANS CONTINUED FROM PAGE 13

the number of bytes to read. If and EOF condition is detected by the read routine before all DE bytes are read a normal (CALL+2) return, not an EOF (CALL+1) return, should be taken with the number of bytes read returned in HL. On the next read entry taken by PTDOS the EOF (CALL+1) return should be taken immediatley with HL=0. If an EOF return is taken by the driver after reading a number of bytes less than DE on a particular call then those bytes will be lost even though HL contained the number of bytes read. The PTDOS documentation on the EOF return is misleading.

* When an entry to a read routine is taken by PTDOS the value in DE (the number of bytes to read) always appears to be the blocksize value in the driver table, no matter how many bytes the executing program requested to read. This may be a PTDOS bug or I have missed some fine point in the documentation.

INSTALLATION

Once the source code for the driver has been written and compiles correctly it need only be EXTRACTed and RETYPEd before it can be used as a driver. A description is provided of the procedure in the comments at the begining of the CUTS example in Appendix C.

For example, if the assembled image driver file is named PRINTER then the commands to make it a device file are:

```
         EXTRACT PRINTER,S
         RETYPE PRINTER,D
```

Its now ready to go. Dont forget to delete the file or change its type back to something else before re-assembling the source (the assembler will try to send the compiled program to the device file).

IN CONCLUSION

This may be more information than you ever wanted to know about device drivers. The best way to put the information into practice is to experiment. Write a driver and play with it. Use it in PTDOS commands where file names are expected and see what happens. I have written device drivers to interface to printers, paper tape readers, an analoge board, a time of day board, and an LSI11. The driver has indeed been a handy software tool.

EXAMPLE DRIVERS

```
*=============================================================*
*          PRINTER DRIVER ROUTINE EXAMPLE                     *
*=============================================================*
;
STAT:    EQU     50H        ;Serial board registers
CMND:    EQU     50H
RREG:    EQU     51H
TREG:    EQU     51H
CONOUT:  EQU     0BC9FH
;
         COPY    EQU$
;
         ORG     8000H      ;Good neighborhood to stick drivers
**********************************************************
DTRB     DW     0                                        *
DTRNB    DW     0                                        *
DTRLB    DW     0                                        *
DTWBR    DW     WRITE       ;Write buffer entry          *
DTWB     DW     WRITE       ;Ditto                       *
DTREW    DW     0                                ***************
DTEOF    DW     EOFWRT      ;EOF entry            DRIVER TABLE
DTCLO    DW     CLOSE       ;CLOSE entry         ***************
DTSEK    DW     0                                        *
DTCTL    DW     0                                        *
DTBLK    DW     1           ;only one byte "BLOCK"       *
DTITO    DB     1           ;imediate transfer on        *
DTINI    DW     INIT        ;initialize                  *
**********************************************************
```

```
*
* INITialize the serial port on the TUART board
*
INIT:    MVI     A,1
         OUT     52H
         XRA     A
         OUT     53H        ;disable interrupts
         MVI     A,04       ;set baud rate to 300
;                           ;and 2 stop bits
         OUT     CMND
         RET
*
*   WRITE - Entry point for writing one block.
*           HL = buffer address to write from
*           DE = number of bytes to write
*
WRITE:   MOV     A,D        ;while DE>0
         ORA     E
         JZ      DONE       ;write is done
;
         MOV     A,M        ;get character
         CALL    PRINT      ;print it
         INX     H
         DCX     D          ;decrement bytes to write
         JMP     WRITE      ;repeat till DE=0
*
*   EOF entry to driver - identical to WRITE
*       except for exit.
*
EOFWRT:  MOV     A,D        ;same routine as WRITE
         ORA     E
         JZ      EOFEXIT
         MOV     A,M
         CALL    PRINT
         INX     H
         DCX     D
         JMP     EOFWRT
*
*  Prints one character on printer
*
PRINT:   PUSH    PSW
PRIN1:   IN      STAT
         ANI     80H
         JZ      PRIN1
         POP     PSW
         OUT     TREG
         RET
*
*
*     EXIT ROUTINES
*
*
DONE:    XTHL               ;normal CALL+2 return
         INX     H          ;all bytes were written normally
         INX     H
         INX     H
         XTHL
EOFEXIT: LXI     H,0        ;EOF CALL+1 return
         RET
*
*   CLOSE entry point.   Line feed up the paper and
*   print end message on console device.
*
*
CLOSE:   MVI     B,20       ;number of LF'S
CLO1:    MVI     A,LF
         CALL    PRINT
         DCR     B
         JNZ     CLO1
;
```

G. EVANS CONTINUED FROM PAGE 14

```
        LXI     H,ENDMSG
        CALL    MSG
        RET
;
MSG:    MOV     A,M        ;print message on console
        ORA     A
        RZ
        CALL    CONOUT
        INX     H
        JMP     MSG
;
ENDMSG: ASC     "PRINTING COMPLETE"
        DB      BELL,BELL,BELL,CR,LF,0
;
        END
```

```
************************************************************
*       PTDRVR  -  Driver for reading from paper tape      *
*                  reader.                                 *
************************************************************
;
      ORG 8000H            ;Good place to stick drivers
************************************************************
DTRB    DW      READ       ;read a block                   *
DTRNB   DW      READ       ;read next block                *
DTRLB   DW      0                                          *
DTWBR   DW      0                                          *
DTWB    DW      0                                          *
DTREW   DW      0                                          *
DTEOF   DW      EOFIT      ;EOF entry          ****************
DTCLO   DW      CLOIT      ;Close entry          DRIVER TABLE
DTSEK   DW      0                              ****************
DTCTL   DW      0                                        *
DTBLK   DW      32         ;"block" size                 *
DTITO   DB      0          ;meaningless for read         *
DTINI   DW      INIT       ;initialize                   *
***********************************************************
*
*       INIT  -  Initialization entry
*
INIT:  XRA     A                 ________\
       OUT     20H                        |
       OUT     21H                        |
       OUT     22H                        |
       MVI     A,0FFH                     |
       OUT     23H                        |
       MVI     A,2EH                      \     Used to initialize
       OUT     20H                         ===  the parallel port
       MVI     A,3EH                      /     connected to reader
       OUT     22H                        |
       XRA     A                          |
       OUT     23H                        |
       IN      21H                        |
       RET                       ________/
*
*       READ - Entry point for reading one block
*              HL = buffer address to read to
*              DE = number of bytes to read
*
READ:   MVI     C,0        ;C = number of bytes read
        IN      0FFH       ;check for "EOF" condition
        ORA     A
        JNZ     EOFEXIT
;
READ1:  MOV     A,D        ;test for zero in DE
```

```
          ORA     E          ;which indicates read done
          JZ      DONE
;
          CALL    GETBYTE    ;get a byte from reader
          JC      DONE       ;"EOF" detected during read
          MOV     M,A        ;store byte
          INX     H
          INR     C          ;increment number of bytes read
          DCX     D          ;decrement bytes to read.
          JMP     READ1
;
GETBYTE:  IN      20H        ;reader byte ready?
          ANI     80H
          JNZ     GETB1      ;yes, go get it.
          IN      0FFH       ;check for "EOF"
          ORA     A
          JZ      GETBYTE    ;no EOF, keep trying
          STC                ;"EOF" detected
          RET
GETB1:    IN      21H
          RET
*
*
*         EXIT ROUTINES
*
DONE:     XTHL               ;normal CALL+2 return
          INX     H          ;all DE bytes were read or some
          INX     H          ;bytes were read prior to EOF.
          INX     H
          XTHL
EOFEXIT:  MVI     H,0        ;EOF CALL+1 return
          MOV     L,C
          RET
*
*
*         EOF and CLOSE entry points
*         These should always exist whether they do
*         anything or not.
*
EOFIT:    NOP                ;nothing to do
          RET
*
CLOIT:    NOP                ;nothing to do here either
          RET
*
          END
```

[EDITOR'S NOTE: Grayson's article above was submitted long ago and was passed back and forth between myself and Tony Severa, who edited the newsletter at the beginning of this year. Sorry for the great delay, but it is still all valid today. The page numbers in the article refer to the original loose-leaf version of the PTDOS manual for version 1.4. The new manual for PTDOS 1.5 has different numbers.]

MICRO COMMUNICATION PROGRAM FOR PTDOS

BY MICHAEL MCKELVEY

December 10, 1980

Stan Sokolow
Proteus
1690 Woodside Road, Suite 219
Redwood City, CA 94061

Dear Stan:

I spoke on the phone today with Bob Jones of Interface Age magazine.

He gave me permission to share my modified version of the Micro Communications Package program that appeared in the March 80 issue. His one concern was that its distribution not result in a profit for anyone. Because of this, we cannot distribute it on a disk that there is a charge for.

We may print a listing of the program in the newsletter and I can record it on someone else's disk for no charge. If people send me a disk with return postage, I will copy the program onto it and send it back. Of course, it would be nice if the people sending me a disk included an interesting program I could copy off.

The communications program operates under PTDOS. It is ORGed at D000 to make full use of 64k memory. Information received is loaded into high memory and when it reaches 64k, it wraps around to 00 and comes up under PTDOS. This information is stored in RAM before being written onto disk. So, the amount of RAM limits the amount of information that can be transfered at a time.

People with less than 64k will have to change the ORG.

You can transmit and receive in both the "S" and "R" modes. In the "R" mode, everything sent back from the host computer is stored in RAM. You can switch from "S" to "R" and back to "S" to control what gets saved. I usually start out in "S" to enter data into a program on the host computer and then switch to "R" with a control-R before the results come back. The ESCAPE key takes you from "R", to "S", to menu and to exit the program.

Information is written to disk, if desired, when you exit the program. It goes into a file called MCP3DATA. You must create this file on unit /0 before using the program.

The disk read and write portions use a "delimited read" or "delimited write" routine where the information in RAM is marked beginning and end with a special character and the transfer continues until that character in encountered. This seemed easier than counting bytes. The WIZ command can be used to transfer the information stored in the PTDOS file MCP3DATA into WordWizard.

Stan Sokolow
December 10, 1980
Page 2

I haven't tried the program to transmit text. So, I don't know for sure if that works. I would be interested to hear of anyone else's experience with that part.

I suggest that anyone wanting to use this program obtain a copy of the March 80 Interface Age article.

If people have questions about it, they may contact me. A call is easier for me to respond to but there's a chance I may not be in and money will be wasted on Ma Bell.

Feel free to contact me again if you have any more questions.

Bob Jones of Interface Age said he would send me a letter confirming the permission. I'll send you a copy when I get it.

In the interest of helping as many Sol users as possible, I have also sent a listing of the program and a similar letter to ASCII. I hope this isn't a problem.

Cordially,

Mike

Michael A. McKelvey

```
                         Micro Communications Package
*                             *** MCP/3 ***
*
* Micro Communications Package
* From March 1980 Interface Age
* Original Program by Dr. Bradford Rehm
* Modified for PTDOS and extended by:
* Michael A. McKelvey
* 330 S. State Street
* Ann Arbor, MI 48104
*
        COPY NPTDEFS
        ORG 0D000H
        XEQ 0D000H
*
****************************************************************
*
*          *** MODE 1 INITIALIZATION & COMMAND LOOP ***
*
* INITIALIZE MCP & GO TO MODE 1
*
STR     LXI SP,STK        set stack pointer
        CALL CRF          send a LF/CR
        LXI H,SON         set msg ptr
        CALL WRT          write it
        LXI H,TPT         set text pointer
        MVI A,CRD         set default CR
        STA DCR           store it
        MVI A,ATS         set default auto-send
        STA AUT           store it
        MVI A,DPB         set default dupl mode
        STA DPL           store it
        MVI A,DMC         set default receive mode char
        STA DMA           store it
        MVI A,DSP         set default speed
        STA BRT           store it
        MVI A,MHI         get high speed bit
        OUT MST           switch modem
```

M.MCKELVEY CONTINUED FROM PAGE 16

```
* MODE 1 COMMAND LOOP
*
LOP     CALL CRF          write a LF/CR
        CALL CRF          and another
        PUSH H            save the text pointer
        LXI H,MNU         load the menu addr
        MVI B,252         load the counter
        CALL L1           WRT w/ bigger count
        POP H             get the text pointer
        CALL CRF          write a LF/CR
        CALL CRF          and another
        CALL CCF          get console char
        ANI 5FH           conv. to UC/NP
        CPI 1BH           is it an 'ESCAPE'?
        JZ EXIT           if yes, exit to PTDOS
        CPI 53H           is it an S?
        JZ SNDT           if yes, MODE 2
        CPI 52H           is it an R?
        JZ RCVT           if yes, MODE 3
        CPI 44H           is it a D?
        JZ CAS            if yes, chg A/S
        CPI 43H           is it a C?
        JZ ACR            if yes, adj EOL char
        CPI 46H           is it an F?
        JZ DPX            if yes, DPX adj
        CPI 57H           is it a W?
        JZ RMM            if yes, RCV M. set
        CPI 5AH           is it a Z?
        JZ DCC            if yes, comm display
        CPI 51H           is it a Q?
        JZ STR            if yes, reinitialize
        CPI 4FH           is it an O?
        JZ RDFILE         if yes, load buffer with outfile
        CPI 42H           is it a B?
        JZ ABR            if yes, adjust baud rate
        CALL SPW          write char
        CALL INV          write error msg
        JMP LOP           loop again
*
* DETERMINE IF TEXT WILL BE SENT CONTINUOUSLY (COMMAND 'C'),
* ON A 'CLR' (COMMAND 'CLR') OR
* AUTOMATICALLY, ON A CLUE FROM THE HOST (COMMAND 'A').
*
CAS     PUSH H            save the text pointer
        LXI H,CSA         load text addr
        CALL WRT          write the string
        POP H             get the text pointer
        CALL CCF          get console char
        ANI 5FH           conv. to UC/NP
        CALL SPW          write it
        CPI 43H           is it a C?
        JZ X1             if yes, load a C
        CPI 0BH           is it a 'CLR'?
        JZ X1             if yes, load a 'CLR'
        CPI 41H           is it an A?
        JZ X1             if yes, load an A
        CALL INV          else, write error msg
        JMP CAS           and try again
X1      STA AUT           store it
        JMP LOP           go to MODE 1
*
* SELECT END-OF-LINE CHARACTER: A DC-3 OR LF/CR
*
ACR     PUSH H            save the text pointer
        LXI H,CRT         load text addr
        CALL WRT          write the string
        POP H             get the text pointer
        CALL CCF          get console char
        ANI 5FH           conv. to UC/NP
        CALL SPW          write it
        CPI 44H           is it a D?
        JNZ A1            else a C?
        MVI A,13H         put a DC-3 in A
        STA DCR           store it
        JMP LOP           go to MODE 1
A1      CPI 43H           is it a C?
        JNZ A2            else, try again
        MVI A,0DH         put a CR in A
        STA DCR           store it
        JMP LOP           go to MODE 1
A2      CALL INV          write error msg
        JMP ACR           try again
*
* DETERMINE WHETHER COMMUNICATIONS WILL
* TAKE PLACE IN HALF- OR FULL-DUPLEX MODE.
*
DPX     PUSH H            save the text pointer
        LXI H,DPT         load the text pointer
        CALL WRT          write the string
        POP H             get the text pointer
        CALL CCF          get the console char
        ANI 5FH           conv. to UC/NP
        CALL SPW          write it
        CPI 48H           is it an H?
        JZ D1             if yes, load an H
        CPI 46H           is it an F?
        JZ D2             if yes, load an F
        CALL INV          else, write an error msg
        JMP DPX           and try again
D1      STA DPL           store it
        JMP LOP           go to MODE 1
*
* STORE THE FDX COMMAND AND SET SEND MODE BYTE
* TO 'AUTO-SEND'.
*
D2      STA DPL           store it
        MVI A,41H         load an A in A
        JMP LOP           go to MODE 1
*
* DETERMINE IF TEXT WILL BE STORED (IN RECEIVE MODE)
* IN 'NORMAL' (CONTROL CHARS EXCLUDED) OR
* IN 'TRANSPARENT' (ALL CHARS STORED) MODE.
*
RMM     PUSH H            save the text pointer
        LXI H,RMT         load the text address
        CALL WRT          write the string
        POP H             get the text pointer
        CALL CCF          get console char
        ANI 5FH           conv. to UC/NP
        CALL SPW          write it
        CPI 54H           is it a T?
        JZ M1             if yes, load a T
        CPI 4EH           is it an N?
        JZ M1             if yes, load an N
        CALL INV          else write error msg
        JMP RMM           and try again
M1      STA DMA           store it
        JMP LOP           go to MODE 1

*
* SELECT BAUD RATE: 300 OR 1200
*
ABR     PUSH H            save the text pointer
        LXI H,SPD         load the text address
        CALL WRT          write the string
        POP H             get the text pointer
```

M.MCKELVEY CONTINUED FROM PAGE 17

```
        CALL CCF         get the console char
        ANI 5FH          conv to UC/NP
        CALL SPW         write it
        CPI 4CH          is it an L?
        JZ B1            if yes, set low speed
        CPI 48H          is it an H?
        JZ B2            if yes, set high speed
        CALL INV         else, write an error msg
        JMP ABR          and try again
B1      STA BRT          store it
        MVI A,SLO        get low speed bit
        OUT SBD          switch computer
        MVI A,MLO        get low speed bit
        OUT MST          switch modem
        JMP LOP          go to Mode 1
B2      STA BRT          store it
        MVI A,SHI        get high speed bit
        OUT SBD          switch computer
        MVI A,MHI        get high speed bit
        OUT MST          switch modem
        JMP LOP          go to Mode 1
*
* DISPLAY THE COMMAND CONFIGURATION.
*
DCC     PUSH H           save the text pointer
        CALL ERA         erase screen
        LXI H,CSA        load the text address
        CALL WRT         write it
        LDA AUT          get auto-send char
        CALL COT         write it
        CALL CRF         write a LF/CR
        LXI H,CRT        load text address
        CALL WRT         write it
        LDA DCR          get EOL char
        CPI 0DH          is it a CR?
        JNZ G1           else, write a D
        MVI A,43H        if yes, load a C
        CALL COT         write it
        JMP G2           then continue
G1      MVI A,44H        load a D
        CALL COT         write it
G2      CALL CRF         write a LF/CR
        LXI H,DPT        load text address
        CALL WRT         write it
        LDA DPL          get duplex char
        CALL COT         write it
        CALL CRF         write a LF/CR
        LXI H,RMT        load the text address
        CALL WRT         write it
        LDA DMA          get the mode char
        CALL COT         write it
        CALL CRF         write a LF/CR
        LXI H,SPD        load text address
        CALL WRT         write it
        LDA BRT          get baud rate char
        CALL COT         write it
        CALL CRF         write a LF/CR
        CALL CRF         and another
        LXI H,PSE        load the text addr
        CALL WRT         write it
        CALL CCF         wait for key
        CALL CRF         write a LF/CR
        POP H            get the text pointer
        JMP LOP          go to MODE 1
```

```
*
* PRINT AN ERROR MESSAGE WHEN THE OPERATOR
* ENTERS AN INVALID CHARACTER IN MODE 1.
*
INV     CALL CRF         write a LF/CR
        PUSH H           save the text pointer
        LXI H,INT        load the text address
        CALL WRT         write the string
        POP H            get the text pointer
        CALL CRF         write a LF/CR
        CALL CRF         one more, please
        RET .            try again
*
* WRITE MODE 1 OPTION PROMPTS
*
WRT     MVI B,20         load the counter
L1      MOV A,M          ready to write
        INX H            increment the pointer
        DCR B            decrement the counter
        PUSH B           save count
        CALL COT         write it
        POP B            restore count
        MOV A,B          ready to count
        CPI 0            is it 0 yet?
        JNZ L1           else, try again
        RET .            else, continue
*
**************************************************************
*
*                *** MODE 2 COMMAND LOOP ***
*
* ISSUE A PROMPT AND GO TO MODE 2
*
SNDT    CALL ERA         erase screen
        MVI A,53H        load prompt in A
        CALL COT         write it
        CALL CRF         write a LF/CR
*
* THE MODEM INPUT LOOP
*
SND     IN MST           get modem status byte
        ANI MSM          save the status bit
        JZ S1            nothing? try console
        IN MOD           if yes, get the input byte
        ANI 7FH          and off the parity
        CALL SPW         write it
        JMP SND          check again
*
* THE MODE 2 COMMAND LOOP
*
S1      CALL CSC         check the console status
        JZ SND           if nothing, check modem
        CALL CCF         get char
        ANI 7FH          and off the parity
        CPI 1BH          is it an 'ESCAPE'?
        JZ LOP           go to MODE 1
        CPI 12H          is it a CTR-R?
        JZ RCVT          go to MODE 3
        PUSH PSW         save A & FLAGS
S2      IN MST           get the modem status
        RAL .            put high bit in carry
        JNC S2           loop until ready
        POP PSW          get A & FLAGS
        CPI 0BH          is it a 'CLR'?
        JZ XMT           send some text
        CPI 01           is it a CTR-A?
        JZ ISO           call LOGON #1
        CPI 02H          is it a CTR-B?
```

M.MCKELVEY CONTINUED FROM PAGE 18

```
        JZ TSO              call LOGON #2
        CPI 0DH             is it a CR?
        CZ DEF              if yes, call EOL byte
        OUT MOD             send console char
        CALL SPC            write it
        JMP SND             check the modem again
*
* SEND BUFFER ON COMMAND IN MODE 2
*
* FIRST, CHECK TO SEE IF WE'RE AT THE BEGINNING OF THE
* BUFFER. IF YES, SKIP THE INITIAL PAD CHAR.
*
XMT     MOV A,H             put contents of H in A
        ADD L               add the contents of L to A
        CPI 1               do they total 1?
        JNZ Y1              else, continue
        INR L               else, increment L
*
* THE BASIC TRANSMIT LOOP: MODEM AND CONSOLE STATUS BYTES
* ARE CHECKED TO PROVIDE EXITS WHEN NECESSARY.
*
Y1      IN MST              get modem status byte
        ANI MSM             get input status bit
        JZ Y2               nothing? try console
        IN MOD              if yes, get the input byte
        ANI 7FH             and off the parity
        CALL SPW            write it
        JMP Y1              check again
Y2      CALL CSC            check console status
        JNZ SND             if data, check it
        IN MST              get modem status byte
        RAL .               put high bit in carry
        JNC Y1              loop until ready
        MOV A,M             get char
        INX H               increment the pointer
        CPI 0FFH            is it a pad char?
        JZ EOT              yes? write msg & exit
        CPI 0DH             is it a CR?
        JZ CRO              send EOL char
        OUT MOD             send char
        CALL SPC            write it
        JMP Y1              get the next char
*
* SEND A DC-3 OR A CR, THEN OBEY ATS COMMAND.
*
CRO     LDA DCR             get EOL char
        CPI 0DH             is it a CR?
        JNZ C1              else, send a DC-3
        OUT MOD             send a CR
        LDA DPL             get duplex mode char
        CPI 46H             is it an F?
        JZ C2               yes? check ATS command
        CALL CRF            else, write a LF/CR
        JMP C2              then check ATS command
C1      OUT MOD             send it
*
* DECIDE IF OPTION IS CONTINUOUS SEND, SEND ON
* COMMAND, OR SEND ON CUE
*
C2      LDA AUT             get auto send byte
        CPI 43H             is it a C?
        JZ Y1               yes? continuous send
        CPI 0BH             is it a 'CLR'?
        JZ SND              yes? send on command
```

```
*
* SEND ON EOL CUE.
*
C3      CALL CSC            a way out if the
        JNZ SND             data is garbled
        IN MST              get modem status byte
        ANI MSM             save the status bit
        JZ C3               if nothing, try again
        IN MOD              get data
        ANI 7FH             and off parity
        CALL SPW            write it
        CPI 0DH             is it a CR?
        JZ C4               yes? go to timer
        JMP C3              else, wait some more
*
* WHEN A CR HAS BEEN RECEIVED, THIS TIMING LOOP
* DISPLAYS ANY ADDITIONAL EOL CHARACTERS AND
* RETURNS TO THE 'XMT' LOOP, WHICH SENDS THE
* NEXT LINE
*
C4      LXI B,4000H         load the counter
C5      DCX B               decrement the counter
        IN MST              get the modem status byte
        ANI MSM             save the status bit
        JNZ C6              if data, write it
        MOV A,B             else, ready count
        CPI 0               is it a 0 yet?
        JNZ C5              else, loop again
        JMP Y1              yes? send next line
C6      IN MOD              get input byte
        ANI 7FH             and off parity
        CPI 13H             is it a DC-3?
        JZ C4               if yes, ignore it
        CALL SPW            write it
        JMP C4              loop again
*
* PRINT A MSG AND RETURN TO MODE 2
* COMMAND LOOP WHEN EOT PAD IS FOUND
*
EOT     PUSH H              save the pointer
        CALL CRF            write a LF/CR
        CALL CRF            one more, please
        LXI H,ETM           load msg address
        CALL WRT            write the string
        CALL CRF            write a LF/CR
        POP H               get pointer
        INX H               incremement H
        JMP SND             go to MODE 2
*
* AUTOMATIC LOGON MESSAGES #1 & #2
*
* THE MSG #1 ENTRY POINT
*
ISO     PUSH H              save the text pointer
        LXI H,IST           load the text address
        CALL I1             send it
        POP H               get the pointer
        JMP SND             go to MODE 2
*
* THE MSG #2 ENTRY POINT
*
TSO     PUSH H              save the text pointer
        LXI H,TST           load the text address
        CALL I1             send it
        POP H               get the pointer
        JMP SND             go to MODE 2
```

M.MCKELVEY CONTINUED FROM PAGE 19

```
I1      MVI B,22          set the counter
I2      IN MST            get the modem status byte
        RAL .             put high bit in carry
        JNC I2            loop until ready
        MOV A,M           get char
        INX H             increment the pointer
        DCR B             decrement the counter
        OUT MOD           send the char
        CALL SPC          write it
        MOV A,B           ready to count
        CPI 0             is it 0 yet?
        JNZ I2            else, try again
        RET .             head for MODE 2
*
***************************************************************
*
*                *** MODE 3 COMMAND LOOP ***
*
* WRITE A PROMPT AND ENTER MODE 3.
*
RCVT    CALL ERA          erase screen
        MVI A,52H         load prompt in A
        CALL COT          write it
        CALL CRF          write a LF/CR
*
* THE MODE 3 COMMAND LOOP
*
        MVI M,0FFH        put text pad in buffer
        INX H             increment the pointer
R1      IN MST            get the modem status
        ANI MSM           get status bit
        JZ R2             nothing? check console
        IN MOD            get modem byte
        ANI 7FH           and off the parity
*
* CHECK RECEIVE MODE: NORMAL OR TRANSPARENT?
*
        PUSH PSW          save the char
        LDA DMA           get default mode char
        CPI 54H           is it a T?
        JZ R3             yes? continue transparent
*
* IMPLEMENT NORMAL MODE OF TEXT STORAGE
*
        POP PSW           get the char
        CPI 0DH           is it a CR?
        JZ R4             if yes, continue
        CPI 0AH           is it a LF?
        JZ R4             if yes, continue
        CPI 09H           is it a HT?
        JZ R4             if yes, continue
        CPI 07H           is it a bell?
        JZ R4             if yes, continue
        SUI 20H           is it a CTR char?
        JM R1             if yes, ignore it
        ADI 20H           put the 20 back
        SUI 7FH           subtract high
        JP R1             if yes, ignore it
        ADI 7FH           put it back
        JMP R4            continue
R3      POP PSW           get the char
R4      MOV M,A           put it in the buffer
        CALL SPW          write it
        INX H             increment the pointer
```

```
R2      CALL CSC          get console status
        JZ R1             nothing? check modem
        CALL CCF          fetch console character
        ANI 7FH           and off parity
        CPI 1BH           is it an 'ESCAPE'?
        JZ R6             if yes, go to MODE 2
        PUSH PSW          save A and flags
R5      IN MST            get the modem output status
        RAL .             put high bit in carry
        JNC R5            loop if not ready
        POP PSW           get A & flags
        CPI 0DH           is it a CR?
        CZ DEF            yes? call EOL routine
        OUT MOD           send it
        CALL SPC          write it
        JMP R1            check the modem
*
* PUT PAD AT END OF BUFFER
* AND GO TO MODE 2
*
R6      MVI M,0FFH        insert buffer pad char
        INX H             increment pointer

        JMP SNDT          go to MODE 2
*
***************************************************************
*
*                   *** MCP UTILITIES ***
*
* GET END-OF-LINE CHAR AND RETURN IT
*
DEF     LDA DCR           get EOL char
        RET               return
*
* CHECK TO FIND WHETHER WE'RE IN HALF- OR FULL-DUPLEX
* MODE. IF IN 'FULL', WAIT FOR CHAR TO RETURN FROM HOST
* BEFORE PROCEEDING.
*
SPC     PUSH PSW          save the char
        LDA DPL           get duplex mode char
        CPI 48H           is it an H?
        JNZ P1            else, go to timer
        POP PSW           if yes, get the char
        JMP SPW           write it
P1      IN MST            get modem status byte
        ANI MSM           save status bit
        JNZ SPV           if data, write it
        CALL CSC          check console status
        JNZ P2    (n.chg) if data, get it
        JMP P1            else, loop again
P2      POP PSW           adjust the stack
        RET .             get the data
*
* CARRIAGE RETURN/LINE FEED
*
CRF     PUSH PSW          save A & flags
        MVI A,0DH         put CR in A
        CALL COT          write it
        MVI A,0AH         put LF in A
        CALL COT          write it
        POP PSW           get A & flags
        RET .             return
*
* ERASE SCREEN
*
ERA     PUSH PSW          save A & flags
        MVI B,16          load counter
E1      MVI A,0AH         put LF in A
```

M.MCKELVEY CONTINUED FROM PAGE 20

```
        PUSH B            save count
        CALL COT          write it
        POP B             restore count
        DCR B             decrement count
        MOV A,B           ready to count
        CPI 0             is it 0 yet?
        JNZ E1            no, loop again
        POP PSW           get A & flags
        RET .             continue
*
* PREPARE TO WRITE THE MODEM CHAR
*
SPV     POP PSW           adjust the stack
        IN MOD            get the modem byte
        ANI 7FH           and off parity
*
* WRITE CHAR FROM
* MODEM OR KEYBOARD
*
SPW     PUSH PSW          save the char
        PUSH B            save count in B
        CPI 11H           is it a DC-1?
        JZ W1             if yes, don't write
        CPI 13H           is it a DC-3?
        JZ W1             if yes, don't write
        CALL COT          else, write it
W1      POP B             restore count in B
        POP PSW           get char
        RET .             back to work
*
***************************************************************
*
*                *** READ FILE INTO BUFFER ***
*
RDFILE  LXI H,NAME        set msg pointer
        CALL WRT          write it
        CALL CRF          write a LF/CR
        LXI H,OFILE       set pointer to filename stg
F1      CALL CCF          get console char
        CPI 0DH           is it a CR?
        JZ F2             if yes, continue
        CALL SPW          write it
        MOV M,A           store it
        INX H             increment the pointer
        JMP F1            go back for next char
F2      LXI D,OFILE       load D with filename pointer
        LXI H,0           load H with buffering parameter
        CALL SYS          call to system
        DB OPEOP          open file
        JMP RDFILE        if error, try again
        LXI H,TPT         load H with text pointer
        MVI M,0FFH        put text pad @ start of buffer
        LXI D,TPT+1       load D with text pointer
        LXI B,19000H-TPT  load B with max # bytes to transfer
        PUSH PSW          save A & flags
        CALL SYS          call to system
        DB RBLOP          read file into buffer
        NOP
        NOP
        NOP
        POP PSW           get A & flags
        CALL SYS          call to system
        DB CLOOP          close file
        NOP
        NOP
        NOP
```

```
        XCHG .            put pointer in H
        MVI M,0FFH        put text pad @ end of buffer
        LXI H,TPT         reset text pointer
        JMP LOP           go to MODE 1
*
***************************************************************
*
*                *** WRITE TO FILE & EXIT ***
*
EXIT    LXI H,FIL         set msg pointer
        CALL WRT          write it
        CALL CRF          write a LF/CR
        CALL CCF          get console char
        ANI 5FH           conv. to UC/NP
        CPI 59H           is it a Y?
        JNZ PTDOS         no?, return to PTDOS
        LXI D,FNAME       load D with file name pointer
        LXI H,0           load H with buffering parameter
        CALL SYS          call to system
        DB OPEOP          open file
        NOP
        NOP
        NOP
        LXI D,TPT+1       load D with text pointer
        LXI B,19000H-TPT  load B with max # bytes to transfer
        MVI L,0FFH        load L with delimiter
        CALL SYS          call to system
        DB DWROP          delimited write to file
        NOP
        NOP
        NOP
        CALL SYS          call to system
        DB EOFOP          end file
        NOP
        NOP
        NOP
        CALL SYS          call to system
        DB CLOOP          close file
        NOP
        NOP
        NOP
PTDOS   CALL SYS          call to system
        DB RETOP          return to PTDOS
*
***************************************************************
*
*                    *** TEXT STORAGE ***
*
SON     ASC '       MICROCOM/3   '
*
MNU     ASC '   COMMAND        DEFAULT'
        DB 0DH
        DB 0AH
        ASC '-------------------------'
        DB 0DH
        DB 0AH
        ASC 'S  SEND DATA'
        DB 0DH
        DB 0AH
        ASC 'R  RECEIVE DATA'
        DB 0DH
        DB 0AH
        ASC 'D  SEND MODE      AUTO'
        DB 0DH
        DB 0AH
        ASC 'W  RECEIVE MODE   NORMAL'
        DB 0DH
```

M.MCKELVEY CONTINUED FROM PAGE 21

```
         DB 0AH
         ASC 'C  EOL CHARACTER  DC-3'
         DB 0DH
         DB 0AH
         ASC 'F  DUPLEX MODE    HALF'
         DB 0DH
         DB 0AH
         ASC 'B  BAUD RATE      1200'
         DB 0DH
         DB 0AH
         ASC 'O  LOAD OUT FILE'
         DB 0DH
         DB 0AH
         ASC 'Q  REINITIALIZE'
         DB 0DH
         DB 0AH
         ASC 'Z  CURRENT'
*
FIL      ASC 'WRITE TO DISK?       '
*
CSA      ASC 'SEND MODE (C,CLR,A):'
*
CRT      ASC 'EOL CHAR (CR,DC-3): '
*
INT      ASC 'INVALID OPERATOR     '
*
DPT      ASC 'DUPLEX MODE (F,H):   '
*
RMT      ASC 'RECEIVE MODE (N,T): '
*
SPD      ASC 'BAUD RATE (L,H):     '
*
ETM      ASC '*** EOT ***          '
*
PSE      ASC 'press any key        '
*
NAME     ASC 'OUT FILE NAME?       '
*
OFILE    ASCZ '0000000000'
*
FNAME    ASCZ 'MCP3DATA'
*
* LOGON MESSAGE #1
*
IST      ASC '[redacted]'
         DB 0DH
         DB 0AH
         ASC '[redacted]      '
         DB 13H
*
* LOGON MESSAGE #2
*
TST      ASC '[redacted]              '
         DB 13H

*
*****************************************************************
*
*                     *** GLOBAL EQUATES ***
*
CSC EQU CONTST             Cons Status Check (active low)
CCF EQU CONIN              Cons Char Fetch
COT EQU CONOUT             Cons OutpuT
MOD EQU 0F9H               MODem data port
MST EQU 0F8H               Modem STatus port
SBD EQU 0FAH               Speed relay port
SHI EQU 00H                Sol HI speed bit
SLO EQU 40H                Sol LO speed bit
MHI EQU 10H                Modem HI speed bit
MLO EQU 00H                Modem LO speed bit
MSM EQU 40H                Modem input Status Mask (active high)
CRD EQU 13H                default EOL byte (DC-3)
ATS EQU 41H                default AuTo-Send byte (auto)
DPB EQU 48H                default DuPlex mode Byte (half)
DMC EQU 4EH                default RCV mode byte (normal)
DSP EQU 48H                Default SPeed byte
VST EQU TST+22             define Variable STorage area
DMA EQU VST                RCV mode byte addr
DPL EQU VST+1              DuPLex mode byte addr
AUT EQU VST+2              AUTo-send byte addr
DCR EQU VST+3              EOL char addr
BRT EQU VST+4              Baud RaTe addr
TPT EQU VST+5              Text PointTer
STK EQU 0CA00H             STacK Pointer
*
*****************************************************************
*END OF PROGRAM
*
```

UNCLASSIFIED ADS

FOR SALE:
1   SOL 20 with 48K plus monitor and cassette
1   OP-80A paper reader
1   Micropolis 5 1/4" disk drive mod 1041
1   Tarbell floppydisk dual drive
1   Extel dot matrix printer plus 2 rolls extra paper
2   Joysticks
Also CP/M on disk
All basic Solus software plus much extended software.
All manuals
3 Verbatim boxes of 5 1/4" disks
Call for complete list. Would like $2500.00 - or make offer.

Paul Kittle
P.O. Box 1285
Loma Linda, CA 92354

Wanted To Trade..
Is anybody upgrading their floppy disk systems and want additional 8" floppy disk drives? I want a working Tarbell disk interface board or a North Star. I can trade working GSI model 110, 8" floppy disk drives with documentation. These units are completly compatible and interchangable with Shugart 800/801 drives.

Edward Enderle
1 Candlewood Court
Huntington, N.Y. 11743
516 673 8169

FOR SALE: Processor Technology SOL 20 Microcomputer--16k memory, TV/Monitor, cassette recorder...includes many programs: assembler, games...In excellent condition. Price $1000.00 or best offer.
Mario I. Guerrero
1023 Nw 84th Dr.
Coral Springs, FL 33065
H 305-752-6737   W 305-587-2900 EX6225

UNCLASSIFIED ADS

COMPLETE S-100/CPM COMPUTER SYSTEM FOR SALE:
ALTAIR MAINFRAME, 18 SLOTS, 13 CONNECTORS, 8080 CPU CARD
TARBELL SINGLE DENSITY DISK CONTROLLER CARD
2 SIEMENS FD-120 8" S.D./D.D. DISK DRIVES
POWER-1 POWER SUPPLY FOR DRIVES, CABINET FOR 1 DRIVE & P.S.
VDM-1 VIDEO CARD, 16X64, MEMORY MAPPED FOR SPEED
3P+S I/O CARD, 1 SERIAL, 2 PARALLEL IN, 2 PAR OUT, STATUS
48K STATIC MEMORY (1-32K TRACE ELEC., 1-16K SPACE BYTE)
TWO 4K DYNAMIC RAM CARDS, ONE IN USE AS ALS-8 SCRATCHPAD
GPM BOARD W/2K CUTER MONITOR, 1K SCRATCH RAM, ALS-8 IN ROM
GEORGE RISK KEYBOARD, FULL ASCII SET; RCA 12" B&W MONITOR
SOFTWARE COMPATABLE WITH THE SOL COMPUTER

IN SERVICE NOW AS AN OPERATING SYSTEM, WITH SOFTWARE AND DOCUMENTATION. PRICE $3000 FIRM, AND WELL WORTH IT! CALL OR WRITE FOR MORE INFO. LEWIS MOSELEY, JR; 2576 GLENDALE COURT, CONYERS, GA. 30208 (404) 922-5788, 6-10PM EST.

GOT TAPE WOES OR SMALL DISK WOES? UPGRADE YOUR S-100 SYSTEM TO A FULL 2 DRIVE 8" CP/M SYSTEM! FOR SALE:
TARBELL SINGLE DENSITY DISK CONTROLLER CARD
2 SIEMENS FD-120 8" S.D./D.D. DISK DRIVES
POWER-1 POWER SUPPLY FOR DRIVES, CABINET FOR 1 DRIVE & P.S.

IN SERVICE NOW AS AN OPERATING SYSTEM, WITH SOFTWARE AND DOCUMENTATION. PRICE $1500 FIRM, AND WELL WORTH IT! CALL OR WRITE FOR MORE INFO. LEWIS MOSELEY, JR; 2576 GLENDALE COURT, CONYERS, GA. 30208 (404) 922-5788, 6-10PM EST.

Distribuco inc.

30315 UNION CITY BLVD. • UNION CITY, CA 94587 • (415) 489-6850

Foodservice Computer Systems, Inc.

Distribuco/Foodservice Computer Systems, Inc. currently has two (2) Processor Technology Sol-Helios II systems we wish to sell.

I wish to announce that I have an IBM Selectric II (Micro Computer Devices, Selecterm, system 9710) interfaced for the Sol for sale. I need to move to a faster printer. This Selectric is loaded with all the options. It has tractor feed, dual pitch, 1/2 backspace, self-correction, and software for a North Star drive. My asking price is $1,675.

Joe Lancaster
1931 Cedar Ridge Drive,#18
Stockton, CA 95207
Phone(home):(209)957-7018

HELIOS II WANTED Need not be in working order. Please state all particulars such as: revision number of controller, serial numbers, date purchased, condition, etc. I can pickup in either San Francisco or Los Angeles area. Joe Maguire, P.O. Box 3742 DT, Anchorage, AK 99510.

WANTED: I need a SOL keyboard. I am also looking for a second set of Wordwizard keyboard labels. If you can help me locate either of these items, please contact me at:

Richard Bjorndal
P.O. Box 13172
Oakland, CA 94661

I might be interested in an empty SOL if I could locate 48-2108 chips, "H" type, for my extra Extensys RAM board. I have dual Micropolis Mod II drives and two controllers and would appreciate your comments on any possiblilies and advantages of using a second SOL with them independently and in combination.

Charles I. Hansing
4741 Hibiscus Av.
Edina, MN 55435
(612)926-1177

I HAVE THREE PEICES OF SOFTWARE ON ECBASIC THAT I USE A LOT AND WILL SHARE, SWAP WITH OTHERS WHO WILL SEND REASONABLE PROGRAMS IN EXCHANGE. NO GAMES, CAI'S...BUT FINANCE, ANALYSIS, SOME HELP CONVERTING MY ECBASIC TO DISK BASIC, ETC. WILL BRING A DELIGHTFUL PROGRAM FOR SOL-20 OWNERS WITH CUTS AND NO DISK.

WHAT I HAVE IS A DATA BASE PROGRAM WITH FULL SORT, REPORT AND ABILITY TO MATHAMATICALLY INTER RELATE COLUMNS... A HOUSEHOLD INVENTORY AND INSURANCE PROGRAM THAT WON'T SORT BUT WILL GREATLY HELP INSURE AND KEEP TRACK OF ALL YOUR GOODIES, AND A MULTIPLE REGRESSION PROGRAM THAT, WILL RUN, READING DATA FROM TAPE OR RAM, ANY NUMBER OF VARIABLES AND OBSERVATIONS THAT YOUR RAM WILL ALLOW, AND GIVE A FULL PRINT OUT OF RESULTS.

GEORGE C. WARNER
AMERICAN EMBASSY BANGKOK KEG
SAN FRANCISCO APO 96346

INFORMATION WANTED: I'm interested in the Epson MX-80 printer, but it doesn't work with the Solos serial port driver. I know it works with the PTDOS driver "Sol3", so the hardware must be compatible. Does anyone have a custom output driver for Solos that will run the Epson MX-80? Please write or call (collect okay after 5:00 pm Pacific time). John Wallace, 340 Corte Madera, Corte Madera, CA 94925, home phone (415) 924-3036.

WANTED: PTC's disk FORTRAN for the Helios and/or WordWizard. Will pay any reasonable price (with documentation). Can't find anybody who still has it. Grayson Evans, 1243 Manchester Ave. Norfolk, Va. 23508, 804-489-9457.

Keyboard Mod kits for the SOL-20 for sale. $24.95 each, in lots of one, $15.95 each, in lots of 10 bulk-packed to a single address, in lots of 100 kits for $12.00 each.

Mr. Barry A. Watzman
560 Sunset Rd.
Benton harbor, Mich. 49022

EDITOR'S NOTE: These kits are described in Protues News, Vol. 2, Number 2, page 10.

I have a SOL computer with a DC Hayes controller board and Northstar Dos. I am looking for some software which would permit me to tie into one of the information data banks such as Source and then get the data into memory so it can be saved and/or printed out.

David H. Simon
Brentwood Square 11661 San Vicente, Suite 903, Los Angeles, 90049 (213) 820-2606
San Francisco Peninsula: Telephone (408) 746-0911

SOLUSER9 -- NORTH*STAR CP/M USER AREA ROUTINES FOR THE SOL/20 COMPUTER

*** Dr. Jim Byram 27 Bencliffe Circle Newton, MA 02166 ***

I offer for publication in PROTEUS/NEWS "yet another" set of CP/M user area routines for the SOL as well as notes on the A/J 841 and a patch for the ELECTRIC PENCIL. SOLUSER9 started with the minimal routines required for the SOL (about 75 bytes including a printer driver as well as the <CR> trap) and includes all of the "bells and whistles" one could desire without coming into conflict with CP/M application programs.

SOLUSER9 does not conflict with ELECTRIC PENCIL in spite of the fact that the PENCIL uses every control character; however, it cannot be used with WORDSTAR (nor can the LifeBoat routines) because the <CR> trap interferes with WORDSTAR.

SOURCES AND IDEAS

1) Lifeboat Associates. 1979. User area routines for CP/M on N*S double density.
2) Maguire, Joe. 1979. Northstar microdisc input/output routine for SOL computer. Proteus/News 2(3): 18-19.
3) Greenlaw, Richard. 1979. Using Heath H-14 printer with SOL. Proteus/News 2(5): 3-4.
4) McGahee, Fr. Thomas. 1979. Interfacing SOL with a Vista disc. Kilobaud Microcomputing 3(9): 56-66.
5) Miller, Alan R. 1980. Double density for CP/M. Lifeboat's North Star version. Interface Age 5(2): 130-134.
6) Hallen, Rod. 1980. Back-space mod for CP/M and Microsoft BASIC. Kilobaud Microcomputing 4(8): 48-49.

SPECIAL FEATURES

1) MODE SELECT -- CP/M warm boot. Same as CTRL-C.
2) CLEAR -- clears screen and regenerates prompt.
3) LOAD -- exits to SOLOS.
4) @ -- deletes the line typed to console. Same as CTRL-U. (CP/M modification required. See below.)
5) DEL -- as with SOLOS. Destructive backspace of characters on the video display. For two-part characters (control characters) and tabs, delete the line with @.

SPECIAL FEATURES USED ONLY DURING CONSOLE: OUTPUT

1) SPACE BAR -- temporarily halts the console output. Any key (except MODE SELECT or CTRL-C) will resume the output. Same as CTRL-S.
2) Any number 0 to 9 -- changes the rate of console output to the video display. 0=default and fastest scroll. 9=slowest scroll.
3) CTRL-X -- changes the CONOUT: device during output. This allows output to be viewed on the screen, a segment to be dumped to the LIST: device, and a return to the screen for the remainder of the output -- all with no break in the listing.
4) MODE SELECT -- CP/M warm boot. May be used to terminate the output either during the listing or following a SPACE BAR or CTRL-S halt. Same as CTRL-C.
5) DEL (=rubout) -- Aborts the console output during listing and returns control to the CP/M Console Command Processor or to the calling program.
6) Any other key struck during console output will be ignored.

AN ADDITIONAL USE OF THE IOBYTE

The CTRL-X routine (again... which is only active during console output) utilizes the IOBYTE (0003H) to direct output to the video display (byte=00) or to the printer (byte=not 00). This byte may also be poked from a BASIC program to alter the console output device during execution of the program.

MODIFICATION OF CP/M

1) On CONIN: input, the Console Command Processor flags CTRL-U and CTRL-X as line delete characters. In v. 1.44 of CP/M (N*S double density), the poll for these two characters begins at 1AFH above the base of BDOS (13AFH in the SYSGEN position). To substitute @ for CTRL-X a's a line delete key, make the following patch:

```
Original code:
AAAFH   FE 18          CPI   18H       ;CTRL-X ???

Modification:
AAAFH   FE 40          CPI   40H       ;@ ???
```

2) To implement those special features used only during CONSOLE: output, modification of the CP/M "BREAK" routine is required. This routine is called during console output to determine whether a CTRL-S has been typed on the CONIN: device. This routine is located 0DFH above the base of BDOS (12DFH in the SYSGEN position). The original routine has the following characteristics:
a) During console output -- CTRL-S brings the output to a temporary halt. Any other key (e.g., rubout) aborts the output and returns control to the Console Command Processor or to the calling program.
b) Following a CTRL-S halt in output -- CTRL-C warm boots the system. Any other key resumes the output.
The required patch is as follows:

```
Original code:
A9EFH   FE 13          CPI   13H       ;CTRL-S ???
A9F1H   C2 FE A9       JNZ   A9FE      ;Abort if not.
A9F4H   CD 09 B6       CALL  B609      ;CONIN: Waits for char

Modification:
A9EFH   FE 7F          CPI   7FH       ;DEL (rubout) ???
A9F1H   CA FE A9       JZ    A9FE      ;Abort if so.
A9F4H   CD 18 BB       CALL  BB18      ;CONCTRL:
```

ADDENDUM

1) The SOLUSER9 assembly listing was generated using the North*Star (UCSD) PASCAL 8080 assembler.

2) Polling routine for Anderson/Jacobson 841 printer with serial interface. Connect the A/J to the SOL with a "null modem" cable as described by Richard Greenlaw in his article on the Heath H-14 (note -- the protective ground, pin 1, is not used by the A/J). To operate the A/J at 150-1200 baud, configure its serial board to the Data Terminal Ready option (jumper on pin F). Insert the following into the SOLUSER9 LIST: routine prior to CALL SDROT.

```
SERST   .EQU    0F8H    ;Serial Status Port
SDSR    .EQU    2       ;Serial Data Set Ready
WAIT:   IN      SERST   ;Is A/J 841 buffer full?
        ANI     SDSR
        JNZ     WAIT    ;If so, wait until empty.
```

3) Did you buy a copy of the ELECTRIC PENCIL (v. SS-IIA for N*S CP/M) to use with the SOL and a selectric printer only to find that selection of single-spacing (LINE SPACING S=1) in the PRINT SUB-SYSTEM COMMAND TABLE resulted in a printed text with double-spacing? S=2 produced triple spacing, etc.?

With IBM Selectric-based terminals, a <CR> actually produces a <CR> followed by a <LF>. The A/J is programmed to ignore any software <LF> which immediately follows a <CR>. Thus, a <CR><LF> sequence coming into the printer produces only a <CR> followed by a single <LF>. The software <LF> is "eaten" by the printer. The problem was that the routine controlling line termination during printing in my version of the ELECTRIC PENCIL put out a <LF><CR>

JIM BYRAM CONTINUED FROM PAGE 24

sequence. The printer puts out the software <LF><CR> and follows with its own <LF> -- the result being unwanted and unexpected double-spacing and the inability to produce single-spaced documents with ELECTRIC PENCIL. The patch to fix this problem is as follows:

Address	Current Data	Change to:	New Data
1414H	0A		0D
1419H	0D		0A

4) The DEL protocol in the CONOUT: routine of SOLUSER9 will fail if you attempt to delete a backslash, as do all of the other published corrections for this problem. With CP/M, delete the entire command line with @. When using either CP/M or MBASIC, if you inadvertantly attempt to delete a backslash, return control to the command mode by typing a RETURN.

SOLUSER9 FILE:SYSTEM.WRK.TEXT

```
0000|                         .PROC   SOLUSER9
Current memory available:    5048
0000|
0000|                   ; * N*S CP/M User Area Routines for SOL/20 *
0000|
0000|                   ; * System constants *
0000|
0000| 002E              MSIZE   .EQU    46      ;Decimal Size of System
0000| A900              BDOS    .EQU    0A900H
0000| B600              BIOS    .EQU    0B600H
0000| BAFF              MODEBYTE.EQU    0BAFFH
0000| BB00              USER    .EQU    0BB00H
0000| BC00              BUFFER  .EQU    0BC00H
0000| BAFC              BUFFWORD.EQU    0BAFCH  ;Sets buffer location
0000|
0000|                           .ORG    0BB00H
BB00|
BB00|                   ; * Define Constants *
BB00|
BB00| BB00              SOLJPT  .EQU    $       ;SOLUSER Jump Table
BB00| C004              SOLOS   .EQU    0C004H  ;SOLOS Warm Entry Point
BB00| C02E              KSTAT   .EQU    0C02EH  ;Keyboard Entry Driver
BB00| C04A              SDROT   .EQU    0C04AH  ;Serial Data Output
BB00| C054              VDMOT   .EQU    0C054H  ;Video Display Routines
BB00| C2E6              PROUT   .EQU    0C2E6H  ;Parallel Output Handler
BB00|
BB00| 0003              IOBYTE  .EQU    3       ;Current CONOUT: Device
BB00| C80B              SPEED   .EQU    0C80BH  ;Speed Control Byte
BB00| 00FA              STAPT   .EQU    0FAH    ;Status Port General
BB00| 0001              KDR     .EQU    1       ;Keyboard Data Ready
BB00| 001A              EOF     .EQU    01AH    ;End-of-File code
BB00| 00FF              TRUE    .EQU    0FFH
BB00|
BB00|                   ; * Jump Table *
BB00|
BB00| C3 ****                   JMP     CINIT
BB03| C3 ****                   JMP     CONST
BB06| C3 ****                   JMP     CONIN
BB09| C3 ****                   JMP     CONOUT
BB0C| C3 ****                   JMP     LIST
BB0F| C3 ****                   JMP     PUNCH
BB12| C3 ****                   JMP     READER
BB15| C3 ****                   JMP     PRSTAT
BB18| C3 ****                   JMP     CONCTRL
BB1B|
BB1B|                   ; * Console initialization *
BB1B|
BB01* 1BBB
```

```
BB1B| AF            CINIT:  XRA     A
BB1C| 32 0300               STA     IOBYTE
BB1F| C9                    RET
BB20|
BB20|               ; * Determine if console input ready *
BB20|
BB04* 20BB
BB20| DB FA         CONST:  IN      STAPT   ;Read port
BB22| 2F                    CMA
BB23| E6 01                 ANI     KDR
BB25| C8                    RZ              ;Not ready
BB26| 3E FF                 MVI     A, TRUE
BB28| C9                    RET             ;Input ready
BB29|
BB29|               ; * Console input *
BB29|
BB07* 29BB
BB29| CD 2EC0       CONIN:  CALL    KSTAT
BB2C| CA 29BB               JZ      CONIN
BB2F| FE 80                 CPI     80H     ;MODE SELECT ???
BB31| C2 ****               JNZ     NEXT
BB34| 3E 03                 MVI     A,03H   ;Sends CTRL-C to CP/M
BB32* 36BB
BB36| FE 8B         NEXT:   CPI     8BH     ;CLEAR ???
BB38| CA ****               JZ      CLEAR
BB3B| FE 8C                 CPI     8CH     ;LOAD ???
BB3D| CA 04C0               JZ      SOLOS   ;Exits to SOLOS
BB40| E6 7F                 ANI     7FH     ;Strip parity
BB42| FE 7F                 CPI     7FH     ;DEL ???
BB44| C0                    RNZ
BB45| 32 ****               STA     DELFLAG
BB48| C9                    RET
BB49|
BB49|               ; * Routine to clear screen *
BB49|
BB39* 49BB
BB49| C5            CLEAR:  PUSH    B
BB4A| 06 0B                 MVI     B,0BH   ;Clears screen
BB4C| CD 54C0               CALL    VDMOT
BB4F| C1                    POP     B
BB50| 3E 0D                 MVI     A,0DH   ;Regenerates prompt
BB52| C9                    RET
BB53|
BB53|               ; * Console output *
BB53|
BB46* 53BB
BB53| 00            DELFLAG .BYTE   0
BB54| 00            BKSLASH .BYTE   0
BB55| 00            LASTCHAR.BYTE   0
BB56|
BB0A* 56BB
BB56| C5            CONOUT: PUSH    B
BB57| 3A 53BB               LDA     DELFLAG ;Routine to translate a
BB5A| B7                    ORA     A       ;DEL (7FH) from the
BB5B| C2 ****               JNZ     DELETE1 ;keyboard into an actual
BB5E| 3A 54BB               LDA     BKSLASH ;destructive backspace on
BB61| B7                    ORA     A       ;the video display.
BB62| CA ****               JZ      RETCHK  ;This routine works under
BB65| AF                    XRA     A       ;two circumstances --
BB66| 32 54BB               STA     BKSLASH ;First, when the DEL
BB69| C3 ****               JMP     ENDOUT  ;results in an <echoed
BB5C* 6CBB
BB6C| 79            DELETE1:MOV     A,C     ;char> as produced by CP/M
BB6D| FE 5C                 CPI     5CH     ;or second, when the DEL
BB6F| C2 ****               JNZ     DELETE2 ;results in a <<backslash>
BB72| 32 54BB               STA     BKSLASH ;<echoed char><backslash>>
BB75| C3 ****               JMP     ENDOUT  ;as produced by Microsoft
BB70* 78BB
BB78| 06 5F         DELETE2:MVI     B,5FH   ;BASIC.
```

JIM BYRAM CONTINUED FROM PAGE 25

```
BB7A| AF                       XRA     A
BB7B| 32 53BB                  STA     DELFLAG
BB7E| C3 ****                  JMP     OUTPUT
BB63* 81BB
BB81| 41             RETCHK:  MOV     B,C      ;SOLOS expects char in B
BB82| 3A 55BB                  LDA     LASTCHAR
BB85| FE 0D                    CPI     0DH      ;CARRIAGE RETURN ???
BB87| C2 ****                  JNZ     OUTPUT
BB8A| B9                       CMP     C
BB8B| CA ****                  JZ      ENDOUT   ;Don't put out two <CR>s
BB88* 8EBB
BB7F* 8EBB
BB8E| CD 54C0        OUTPUT:  CALL    VDMOT
BB8C* 91BB
BB76* 91BB
BB6A* 91BB
BB91| C1             ENDOUT:  POP     B
BB92| 79                       MOV     A,C
BB93| 32 55BB                  STA     LASTCHAR
BB96| 3A 0300                  LDA     IOBYTE   ;Allows BASIC program
BB99| B7                       ORA     A        ;to determine output device
BB9A| C2 ****                  JNZ     LIST     ;by poking value to IOBYTE
BB9D| 79                       MOV     A,C      ;CP/M expects char in A
BB9E| C9                       RET
BB9F|
BB9F|                ; * Routine to control console output *
BB9F|
BB19* 9FBB
BB9F| FE 18          CONCTRL:CPI      18H      ;CTRL-X ???
BBA1| CA ****                  JZ      XCHANGE
BBA4| FE 13                    CPI     13H      ;CTRL-S ???
BBA6| CA ****                  JZ      PAUSE    ;The SPACE BAR may also
BBA9| FE 20                    CPI     20H      ;be used to temporarily
BBAB| C2 ****                  JNZ     SCROLL   ;halt console output.
BBA7* AEBB
BBAE| CD 29BB        PAUSE:   CALL    CONIN    ;Any key resumes the
BBB1| C9                       RET              ;output.
BBAC* B2BB
BBB2| FE 3A          SCROLL:  CPI     3AH      ;This routine may be used
BBB4| D0                       RNC              ;to regulate the rate of
BBB5| FE 30                    CPI     30H      ;console output.  During
BBB7| D8                       RC               ;output, enter any number
BBB8| E6 0F                    ANI     0FH      ;between 0 (fastest) and
BBBA| 17                       RAL              ;9 (slowest) to change
BBBB| 17                       RAL              ;rate.  Default=0.
BBBC| 00                       NOP              ;Add another RAL for more
BBBD| 32 0BC8                  STA     SPEED    ;delay.
BBC0| C9                       RET
BBC1|
BBC1|                ; * Routine to change console output device *
BBC1|
BBA2* C1BB
BBC1| 3A 0300        XCHANGE:LDA      IOBYTE   ;The CONOUT: device may
BBC4| B7                       ORA     A        ;be changed during output
BBC5| CA ****                  JZ      INVERT   ;by typing a CTRL-X.
BBC8| AF                       XRA     A        ;Thus, a partial listing
BBC9| C3 ****                  JMP     ENDXCH   ;may be directed to the
BBC6* CCBB
BBCC| 2F             INVERT:  CMA               ;printer.
BBCA* CDBB
BBCD| 32 0300        ENDXCH:  STA     IOBYTE
BBD0| C9                       RET
BBD1|
BBD1|                ; * Printer on-line *
BBD1|
BB16* D1BB
BBD1| AF             PRSTAT:  XRA     A
BBD2| C9                       RET
BBD3|
BBD3|                ; * Printer driver *
BBD3|
BB9B* D3BB
BB0D* D3BB
BBD3| C5             LIST:    PUSH    B
BBD4| 41                       MOV     B,C      ;SOLOS expects char in B
BBD5| CD 4AC0                  CALL    SDROT    ;For a parallel printer,
BBD8| C1                       POP     B        ;change to PROUT (C2E6H).
BBD9| C9                       RET              ;Returns with char in A
BBDA|
BBDA|                ; * Reserved *
BBDA|
BB13* DABB
BBDA| 3E 1A          READER:  MVI     A, EOF
BBDC| C9                       RET
BBDD|
BBDD|                ; * Reserved *
BBDD|
BB10* DDBB
BBDD| C9             PUNCH:   RET
BBDE|
BBDE|                          .END
```

Dr. J.E. Byram
27 Bencliffe Circle
Newton, MA 02166

RANDOMIZE FOR PT BASIC

Have you ever noticed that every time you load a BASIC game, the random number generator starts in the same place? Have you ever noticed that some BASICs have a randomize command, but PT BASIC (cassette or disk) does not? I have had this problem, and have solved it. The usual answer is to ask for a seed number at the beginning of the program, but this also causes repetition. I also feel that this makes the program look messy and unprofessional. The real answer is to cause the random number generator to increment a random number of times. So far, so good; but if we couldn't get the first random number, how can we get the second? Most games start by asking for the player's name. If we were to increment the random number generator until the player responded with his/her name, we would start with a different random number every time. (It has been my experience that very few people take exactly the same number of micro-seconds between starting a program, and entering their name.) However, PT BASIC does not allow us to use the WAIT statement for the keyboard port. So how do we know when the player has started typing his/her name? We ask the port directly:

```
  60 REM- THE FOLLOWING ROUTINE RANDOMIZES
  70 PRINT "WHAT IS YOUR NAME? ";
  80 LET C=INP(252)
  90 LET L=INP(252)
 100 LET R=RND(0)
 110 IF C=L THEN 90
 120 PRINT CHR(L);
 130 INPUT "",N$
 140 LET N$=CHRL(L)+N$
```

Try it, you'll like it!

Richard Bjorndal
P.O. Box 13172
Oakland, CA 94661

PTC_PILOT_TO_DISK

Letter_to_Editor
May_29,_1980

Bob Stek wanted a way to save and load PTC PILOT programs in CP/M [PROTEUS NEWS 2(5):10]. Lewis Moseley, Jr. suggested using the excellent PTC PILOT editor for writing letters [PROTEUS NEWS 2(2): 20]. I have been using PILOT together with a custom printer driver to write voluminous psychiatric reports and have made a quick and dirty interface with CP/M.

If PILOT is simply saved on diskette any associated programs will be lost when it is initialized from location 100H. A simple castration eliminates this quirk forever. XEQ PILOT from cassette to load and initialize it. Set memory to the size desired. I chose to `SET M=18800´. Type the command `INFO´ and write down the memory location given at the top line. This will be needed for saving to disk. Now write PILOT back out to cassette using the command `CUSTOM´. This maneuver modifies PILOT so that it will execute the PILOT program upon initialization rather than destroying it. Read the castrated version of PILOT back into memory. The customized PILOT will execute rather than erase memory when the `SCRATCH´ command is given. To erase memory, key control-L, then MODE SELECT, then RETURN. Memory size cannot readily be changed.

Refer to the PILOT memory location that we wrote down. Take the high-order hexadecimal digit, multiply it by 16 then add the second digit. My memory location was 4970, so I obtained 4 X 16 + 9 = 73. This is the number of 256-byte pages of information that must be saved. Give CP/M the command `A>SAVE 73 PILOT.COM´ (or whatever your number of pages was) and in 3 seconds you will have a file named PILOT.COM which you can load and execute by typing `PILOT´.

To save a program, construct a PILOT program in the ordinary fashion, then key MODE SELECT to return to the PILOT command mode. Enter a backslash followed by `CALL:0´ to do a warm boot of CP/M. You can now save your program via the CP/M command `A>SAVE 73 NAME.COM´. Entering `NAME´ will then cause CP/M to run your PILOT program. Obviously, each time a program is saved, the entire PILOT interpretor and editor is saved along with it. This is an inefficient use of disk space, but at least it works. The cassette save features of PILOT remain operational.

It is useful to have the first line of a PILOT program read `EDIT´. This causes the program to go into edit mode when it is brought up by CP/M. In the final version wipe out the `EDIT:´ with a control-P so that the program will print out. Before this, keep saving successive snapshots of the program onto disk as backup.

Chaining PILOT programs is easy under CP/M. Create a SUBMIT file listing the names of PILOT programs (without the .COM attribute) in the order they are to be executed and use the CP/M `SUBMIT´ command. Each PILOT program to be chained should end with the command `CALL:0´, which causes a warm boot of CP/M and the invocation of the next PILOT program on the SUBMIT file.

Because a line of PILOT program which has no colon is typed out when the program is run, PILOT lends itself to use as a word processor. Text may simply be typed in. The command `SET:O=1´ anywhere along in the program will divert subsequent lines to a printer attached to the serial interface, and `SET:O=0´ restores output to the video screen.

For fully formatted and right-justified printing I wrote a custom driver for my Teletype-43 which interprets output from PILOT one character at a time. To send the output to this driver I use `SET: O=3´. The PILOT facilities readily permit personalized form letters to be printed. I have certain skeleton reports available on disk which I read in and then tailor for particular purposes. These "boilerplate" forms as well as computer-aided-instruction programs can be easily exchanged with other SOL users via CP/M diskettes.

Many of PTC PILOT´s features rival or outshine those of the Electric Pencil. Lacking are a block move command, a line-copy feature and an insert-macro ability. If a carriage return were automatically entered toward the end of a line the typist would not need to keep looking up at the screen. Below is an input driver routine which handles that task. When editing existing text it may be desirable to disable automatic entry of the carriage return character. This is done by striking the LOAD key. The form-feed character echoed to the screen may be removed with the DEL key.

If a colon is included in the text, PILOT may be confused into interpreting the line of text as a command. This is avoided by starting each new line with a colon. The input driver provides this colon automatically.

The driver may be expanded in a straightforward way to send a stream of characters from memory into the PILOT program being edited. I hope to create this `macro´ feature shortly.

I would be grateful for any suggestions and for PTC PILOT source code (some of which was printed in Dr. Dobbs´) so that I could add some commands and rework the editor into a super word processor.

Sincerely,

Myron Pulier, M. D.
101 Cedar Lane
Teaneck, New Jersey 07666

```
TYPE B:PILUP.PRN


                  *               P I L U P
                  *
                  *V3 29MAY80             COPYRIGHT (C) 1980 MEGASOFT
                  *
                  *PILOT upgrade by Myron Pulier, M. D.
                  *
                  *If cursor is past position 50 a CR is substituted
                  *for any blank typed in Edit mode, obviating the need
                  *to watch the screen while typing text.  This feature
                  *is enabled and disabled by means of the LOAD key.
                  *
                  *In any case, a colon is automatically inserted at
                  *the beginning of each new line.
                  *
                  *PILOT must be modified by `1CF6:00 C9/´ to use this
                  *input driver.
                  *
                  *To install PILUP via CP/M execute
                  *       A>DDT PILUP.HEX
                  *Then, in DDT
                  *       -GO
                  *Next,  A>PILOT
                  *Then alter 1CF6 using SOLOS
                  *******************************************************
                  ;
003A =            COLON   EQU     ´:´       ;COLON
0002 =            CTLB    EQU     2         ;CONTROL-B
000D =            CR      EQU     0DH       ;CARRIAGE RETURN
C02E =            KSTAT   EQU     0C02EH    ;SOLOS KEYBOARD DRIVER
008C =            LOAD    EQU     8CH       ;SOL `LOAD´ KEY
```

MYRON PULIER CONTINUED FROM PAGE 27

```
C808 =          NCHAR   EQU     0C808H  ;CURSOR POSITION
                ;
C900                    ORG     0C900H
                ;
C900 3A53C9     PILUF   LDA     FORMER  ;GET FORMER OUTPUT
C903 FE0D               CPI     CR      ;WAS IT A CR?
C905 CA14C9             JZ      INSERT  ;YES, RETURN A COLON
C908 FE02               CPI     CTLB    ;WAS IT A CTLB?
C90A CA14C9             JZ      INSERT  ;YES, RETURN A COLON
C90D CD1BC9             CALL    BREAK   ;ELSE INPUT KEYED DATA
C910 3253C9             STA     FORMER  ;SAVE IT FOR NEXT TIME
C913 C9                 RET             ;AND RETURN IT IN A
                ;
C914 3D         INSERT  DCR     A       ;RESET Z
C915 3E3A               MVI     A,COLON ;GET A COLON
C917 3253C9             STA     FORMER  ;SAVE IT FOR NEXT TIME
C91A C9                 RET             ;AND RETURN IT IN A
                ;
C91B CD2EC0     BREAK   CALL    KSTAT   ;GET CHARACTER
C91E CA1BC9             JZ      BREAK
C921 FE8C               CPI     LOAD    ;IS IT LOAD?
C923 CA40C9             JZ      FLIP    ;YES, FLIP TOGGLE
C926 FE20               CPI     ' '     ;IS IT BLANK?
C928 C0                 RNZ             ;NO, RETURN WITH IT
C929 3A08C8             LDA     NCHAR   ;GET CURSOR POSN
C92C FE32               CPI     50      ;PAST 50?
C92E DA3DC9             JC      SPACE   ;NO, RETURN SPACE
C931 3A54C9             LDA     TOGGLE  ;TOGGLE SET?
C934 FE8C               CPI     LOAD
C936 C23DC9             JNZ     SPACE   ;NO, RETURN SPACE
C939 3D                 DCR     A       ;YES, RESET Z FOR RETURN
C93A 3E0D               MVI     A,CR    ;GET CR FOR RETURN
C93C C9                 RET
                ;
C93D 3E20       SPACE   MVI     A,' '   ;GET SPACE FOR RETURN
C93F C9                 RET
                ;
C940 3A54C9     FLIP    LDA     TOGGLE  ;IS TOGGLE SET?
C943 FE8C               CPI     LOAD
C945 CA4DC9             JZ      RESET   ;YES, RESET TOGGLE
C948 3D                 DCR     A       ;NO, GET LOAD CHAR
C949 3254C9             STA     TOGGLE  ;SET TOGGLE
C94C C9                 RET             ;RETURN WITH LOAD CHAR
                ;
C94D 3C         RESET   INR     A       ;CREATE A SET-CODE
C94E 3254C9             STA     TOGGLE  ;SET TOGGLE WITH IT
C951 3D                 DCR     A       ;RESTORE LOAD CHAR, RESET Z
C952 C9                 RET             ;RETURN WITH LOAD CHAR
                ;
C953 00         FORMER  DB      0       ;PREVIOUS CHARACTER
C954 8C         TOGGLE  DB      LOAD    ;FLIP-FLOP ENABLED
                ;
C955                    END

A>
```

BOSTON AREA ANYONE?

Edwin Meyer wants to know if any Boston area members would like to form a local group for Sol owners. If you are interested, please contact Edwin at 339 Newbury Street, Boston, MA 02115.

...ON HELIOS IN A Z80 SYSTEM

NOV 16,1979

Dear Stan,

I recently recieved Volume 2 of Proteus/News and was very pleased with the content and quality of your newsletter. You have done a very good job. Keep up the good work.
I'm not sure how I didn't know it existed for so long but I didn't! I saw one copy of Access in a dealers store and figured that was it. I only read about Proteus in a letter in Dr. Dobbs recently. You should advertise in a few of the Journals.

Inclosed you will find my check and an order for your H series disks. Also Inclosed you will find this disk with a number of programs you are welcome to. This was an easy lot to assemble on short notice. I have many more but they will take a little cleaning up and documentation.

I am running PTDOS with the Helios (I bought one of the first kits at $1800.00) for a few years. I dont have a Sol but put together an S100 system from scratch using a Z80 CPU board. I started out running CUTER and still use it from time to time.

I have a large library of programs that I have converted to run on the Helios either in EDBASIC or in Z80 assembly language. I have converted the TDL Macro Assembler and Text Processor. I use the Macro Assembler almost for everything. I have converted and running the Z80-ZAP Disassembler that appeared in the May 79 Dr. Dobb's, WHATSIT, Stock analysis programs, a 16 bit math library (with string functions) and numerous other programs.
I have developed various games, drivers, and application programs but they are writter in Z80 assembly language but I dont't know how usefull they would be to your members. If you are interested let me know and I will send in a few of the better ones with good documentation.

I am also running CP/M from Lifeboat and I'm currently writing an interface program to run programs available from Lifeboat under PTDOS. It is a go-between program that looks like CP/M to running programs but passes control to PTDOS with properly formatted information. I'm doing this because there are a vast library of programs available from Lifeboat (enough to keep me busy for the rest of my life) but I dont like CP/M. Lets face it; PTDOS is probably the best microcomputer operating system written. I have used most of them and many of the mini and large computer operating systems. I've learned to dislike operating systems in general due to thier poor quality. It's a relief to use PTDOS. Until someone comes up with a micro-UNIX I'm sticking with it. (Someone is working on a micro-UNIX by the way and using a Helios to do it.)

I throughly enjoyed the back issues of Proteus/News.
The Helios Parameter Scanner article in the Volume 2 No. 5 issue was excellent. I would like to write a similar article on a general purpose error handling routine.

Keep up the good job.

Sincerely,
Grayson Evans

...ON PER SCI DRIVES, HELIOS SERVICE MANUAL, ETC.

November 9, 1980

Dear Stan:

I intended to write another letter shortly after our telephone conversation. In the meantime, however, Tony organized a repair operation involving the shipment of SOL-HELIOS Systems. His experience could provide some useful information about how well the alignment holds during shipment.

I have suspected for some time that the Model 277 PerSci Disk Drive in the HELIOS sometimes drops out of alignment during handling and replacement in the HELIOS cabinet. There have been others who have noticed drive malfunctions after replacing the drive and tightening the mounting screws that project though the bottom of the HELIOS cabinet into the base of the drive. These suspicions seem to be confirmed by a PerSci Field Change Bulletin issued March 10, 1980.

Field Change #1 states that the positioner rods in the drive are sometimes distorted when the screws that hold them in place are tightened. An alignment, if made while the rods are distorted, will not hold if the rods spring back into position. Apparently this came to PerSci's attention as a result of complaints that some of the drives shipped to customers were out of alignment on arrival. They say that vibration causes the rods to spring back into normal position throwing the drive out of alignment.

PerSci is now offering a kit that prevents the positioner rods from being deformed. I have the kits for my drives, but I have not installed them. One of my drives is out of alignment apparently as a result of transporting it to and from a school demonstration. I intend to install the kit before I align it.

Although I am not really in the service business, I repair and align my own SOL-HELIOS systems, and I have repaired and aligned a number of Systems for others. I do technical writing and have a modern electronic shop to support my writing; and I use my shop equipment, when necessary, for computer repair and alignment.

Most of the SOL-HELIOS System failures I have seen have been HELIOS failures. I consider the Controller and Formatter Boards in the SOL as part of the HELIOS, so the empty SOL has not been a failure problem.

Of the components, positioner lamps fail quite frequently, and when the lamp is replaced, the HELIOS should have at least a partial alignment. I have had one component fail on a Formatter Board, and I know of two or three other component failures. I have put a new delay line on two or three 16K boards, but memory has not been much of a problem otherwise.

Some systems can be put back into operation with an alignment and nothing more. The deformed positioner rods mentioned above can be responsible for some of the mis-alignments, but certainly not all of them. There are voltage, timing, and perhaps other adjustments that are not related to the deformed positioner-rods.

I do not want to over simplify the situation, but I think many of the SOL-HELIOS hardware failures are repetitions of a few causes. I have in mind a few components which fail rather often. These components are not difficult to replace, but the HELIOS may need alignment after some component failures. Poor alignment is a very important cause of either a complete failure or a failure in which the system makes frequent errors.

I have most, possibly all, of the repair documentation. The alignment is covered in the documentation, however, I have found errors and misleading statements in the procedure. The hand-drawn, wave-form illustrations used in the documentation are only rough approximations of the wave forms displayed by the oscilloscope. In other words, the documentation can be improved considerably.

Aside from errors and misleading statements, the hand-drawn wave-forms in the documentation leave doubt as to exactly what the wave must look like on the oscilloscope to represent satisfactory alignment. I remember I wasted a lot of time in the beginning trying to determine just how good a wave had to be to prevent computer malfunction.

Stan, if SOL-HELIOS users have the right equipment and the proper guidance, I am sure that many will be able to do their own minor repairs and alignments or find local help qualified to do the work. The equipment must include a service diskette, and an alignment diskette. Adequate documentation must be available also. In fact, if PT's documentation is used, the problem confronting a person doing an alignment for the first time may be the documentation. The service and alignment diskettes plus an adequate minor repair and alignment manual will make the job fairly easy for one who can follow directions and operate an oscilloscope.

I have the equipment needed to write and prepare a manual for publication. The manual would be illustrated with numerous oscilloscope photographs, equipment photographs, and line drawings. In other words, the manual would contain oscilloscope photographs of satisfactory wave-forms rather than hand drawings, and would give step-by-step directions for the entire procedure. It also would contain repair procedures for a few commonplace failures. And, it would be written in non-technical language.

In addition, I have the service and alignment diskettes. If copyright releases can be obtained, it should be possible to provide service and alignment diskettes, when needed, with the repair and alignment manual.

Obviously, such a manual will be expensive to prepare and publish. I cannot do the job without a monetary return. At the moment, I cannot quote a selling price, because I have not estimated the cost of writing and publication -- a rather complicated process. If the manual is published, my first thought is that it should be sold through PROTEUS.

I need to get from you, Stan, and from the membership an idea of the number of manuals that might be sold and of the price members would be willing to pay on a money back guarantee. The cost of the manual should not include service and alignment diskettes, because some may have them. Obviously, if there is little demand, I will not consider the project further. On the other hand, if there is a big demand, I can give the job a high priority and turn it out in a few weeks.

Now a new subject! I am putting CP/M on my SOL-HELIOS System. In addition to output from such as CBasic and Micro-Soft Basic, I want to run WordStar and take advantage of WordStar's proportional spacing. CP/M is running fine, but I need a printer driver so that I can send output to the printer. Any driver that will run is a good start; proportional spacing may not be the driver to start with.

I have the Diablo HyType II Printer, Model 1355WP. This is the Diablo word-processing printer. It uses the 96 character metal print-wheels; it cannot use plastic print-wheels. It is capable of proportional spacing and can use metal proportional-spacing print-wheels -- Cubic PS 96, for example. Metal print wheels which have fewer than 96 characters can be used, but there must be no attempt to print characters not on the print wheel -- in fact, this print out is from a Titan 12 print wheel running on WordWizard.

I am willing to pay for help with a driver which will give me the proportional spacing of WordStar or, perhaps, another word-processing program which has proportional spacing. What I would really like is a revision of WordWizard which has true proportional spacing. I am writing this letter with WordWizard and I like it. The trouble is that I need true proportional spacing in order to give my technical writing the best possible appearance.

If possible, I want to keep this Diablo on the parallel port and use it to output from both PTDOS and CP/M. If it is necessary to run the output from WordStar to a printer through the SOL's serial port, then I will need to buy an additional printer or, perhaps, modify this Diablo. I prefer to buy an additional printer, however, unless it is possible to switch this one from parallel- to serial-port and back as required by what is being run, PTDOS or CP/M.

Sincerely yours,

Hal A. Lindsey

...ON UPGRADE FOR HELIOS

September 15, 1980

Dear Stan,

Enclosed you will find a Helios disk with my LOADM program and associated documentation files for donation to the PROTEUS library. A copyright release is enclosed. Please ship me a copy of the new H-4 library disk.

My consulting business, LMC Engineering, supports Helios systems in the Los Angeles and Orange County area. My principal activities include system engineering, assembly language programming, PTDOS interfacing, and device driver generation. A surprisingly large number of businesses here use Helios systems. I just finished a major effort with a local police department that now uses a four-drive system for arrest and crime information reporting. I frequently find that these users are able to write the application software in EDBASIC but may not use the file system effectively. System menus and automatic program-loading on bootup are also areas that deserve more attention. I recommend PROTEUS membership to all my customers.

I am considering writing an article on a PRACTICAL approach to using CP/M with the Helios drives. I have put several systems together already and am convinced this is the best approach for most users. I use a RELIABLE single-density controller board (and the Persci data separator) to operate the unmodified Helios drives. No other system changes are needed--just plug the Helios cable onto the new controller board and go! Standard soft-sector diskettes are used and are fully interchangeable with other CP/M users. I will offer this special controller board and the required CP/M BIOS for sale if there is sufficient interest.

I have considered the tradeoffs between single-density and double-density controllers. Frankly, the increase in disk capacity is had at a terrible cost--poor reliability. To most of my customers, a system that always comes up and works is far more usable than one with more storage but questionable reliability. One must consider the Persci drive's spindle-speed regulation problem--it is noted for serious ISV (Instantaneous-Speed Variation) problems. This performance causes much trouble at double density. Some trouble is common with single-density controllers which use digital data separators; read/write retries are common but may not be noted by the operator.

I modified a BIOS to alert the console at each retry when I was testing controllers. I found that all retries were eliminated when I started using the analog data separator in the Persci drive. This is the scheme used by the Helios controller, too. These systems have performed very well even in high-usage environments. Further, it is possible to switch back to PTDOS just by moving the disk-drive cable to the Helios controller.

I am certainly interested in some of the source files PROTEUS recently acquired. Your perseverance in obtaining this highly valuable package is greatly appreciated.

Glad to see that you're back with us, Stan! Good luck.

Larry McDavid 185 South Alice Way Anaheim, CA 92806 evenings 714-630-5672

...ON CONVERTING PTDOS FILES TO LIFEBOAT CP/M FILES

THIS PROCEDURE IS TO BE USED IN SOL-HELIOS SYSTEMS HAVING 48K OF MEMORY AND IS DESIGNED TO CONVERT A PTDOS FILE OF UP TO A MAXIMUM UP 8K IN LENGTH TO A CP/M TEXT FILE. THIS IS DONE WITH THE BASIC LANGUAGE PROGRAM CPM-TXT WHICH IS ON THIS DISK. I'M SURE THAT THERE IS AN EASIER WAY TO DO THIS BUT THIS WAY WORKS FOR ME.

STEPS TO BE TAKEN TO CONVERT PROC.TECH. BASIC OR TEXT FILES TO LIFEBOAT CP/M FILES IN A HELIOS.

```
1. BOOT UP HELIOS
2. SET BU=9000
3. CREATE TEXT FILE OR SAVE A VERSION OF YOUR BASIC PROGRAM
   AS TYPE,T
4. ZIP   (I USUALLY USE ZIP 0)
5. READ FILE,5000 (RECORD THE HEX LENGTH OF THE FILE)
6. LOAD BASIC AND CPM-TXT
7. RUN  (DO NOT SCRATCH OR ALL WILL BE LOST)
8. BYE
9. IMAGE FILE,!4C0,7000,>LGTH
10. ZIP
11. READ FILE,100,>LGTH   (THIS PUTS THE FILE AT CP/M LOCATION)
12. REMOVE PTDOS DISK, RESET, INSERT CP/M DISK AND BOOT
13. SAVE n FILE.TXT      (SEE CP/M MANUAL TO DETERMINE n)
```

YOU WILL NOW HAVE A COPY ON CP/M OF ALL BUT THE FIRST LINE OF THE PROGRAM. THE FIRST LINE IS FULL OF GARBAGE AND WILL HAVE TO BE EDITED TO CORRECT.

THE FIRST TWO BYTES OF THE IMAGE FILES CREATED IN STEP 9 DEFINE THE LENGTH OF THE FILE IN BYTES AND ARE THE CAUSE OF THE FIRST LINE GARBAGE. I DON'T KNOW HOW TO AVOID THIS.

J.G.Z. 3/28/79

```
 10 REM*****CPM-TXT
 20 PRINT "SUGGEST 5000H OR 20480D FOR START"
 30 PRINT "AND 7000H OR 28672D FOR NEW FILE"
 40 PRINT "---- TO USE 7000H CHANGE BUFFER TO 9000H"
 50 PRINT "MAKE AN IMAGE FILE STARTING AT 7000H TO RELOAD FOR
 60 PRINT "DECIMAL ADDRESS OF MEMORY WHERE THIS FILE NOW"
 70 INPUT "BEGINS. (IT SHOULD BE 20480 )   : ",X
 80 INPUT "DECIMAL ADDRESS OF END OF THIS FILE : ",Y
 90 LET L=X
110 FOR I=X TO Y
120 LET Q=PEEK(I)
130 POKE L+8192,Q
140 IF Q=13 THEN  GOTO 190
150 LET L=L+1
160 IF I=Y THEN  PRINT "FINISHED": END
170 NEXT I
180 REM**ADD A LINEFEED FOR CP/M
190 POKE L+8193,10
200 REM***ADD A BYTE TO THE NEW FILE TO MAKE ROOM FOR
210 REM***THE CARRIAGE RETURN AND THE LINEFEED BOTH.
220 LET L=L+1
230 GOTO 150
240 REM***CPM-TXT
```

Gib Zeratsky
813 Inlet Road
Green Lake, WI
54941

...ON VDS-1 PROBLEM WITH 8085

9/10/80

Dear Stan:

First, please accept our thanks for taking on the editing job for Proteus again; you have redoubled the debt all members owe you, which was already more than one could expect from any ten yeomen! I do wish I could offer more than the limited time I have available as a resident, and that we can all pitch in to try to help out.

I have a problem which may become more common as more of those with VDM-1's move up to newer cpu's. I hope it is solvable and that someone can help! I have purchased the Godbout Dual Processor board and have discovered that the VDM-1 does not work at all with the 8085 at 5mhz, and occasionally malfunctions at 2mhz (usually with fast machine language.) I called Godbout, and apparently the problem is that the VDM-1 does not meet the IEEE S-100 standards for asserting the XRDY line to stop the processor while it arranges access to the screen memory. The 8085 meets the specs but is less tolerant than the 8080 or Z-80. My VDM-1 works fine with a 4mhz Z-80 or 2mhz 8080. I have not yet been able to try the 8088 (no software yet!) but am looking forward to Pascal/M for the dual board and an 8086/8088 cross assembler.

Does anyone out there know how to handle this? I hope that someone with hardware knowledge can work out a fix - it certainly doesn't matter if it slows things down some! I do love the VDM-1 and would hate to have to buy a terminal!

As others who started with CUTER/Imsai systems have undoubtedly done, I have expanded my system to include the Discus 2D and CPM 2.2. I have been extremely happy with the Discus system, software, and support - including such things as upgrade from CP/M 1.4 to 2.2 for $50. and a very reasonable charge for proms and a full set of software when I moved the disc controller to F800.

I have needed to maintain CUTER for tape data file input as part of my system, and so worked out reassembled versions for different origins and VDM addresses, have the PTC source in CP/M compatable form, and can use the Discus auto feature to boot CP/M, load CUTER, and then use Cuter for IO and to run PT software. This all is necessary because the GPM board with CUTER in rom decodes all the addresses for the rom ALS8 and so you can't use the GPM and the Discus controller at the same time.

I haven't seen it noted that the PTC source for CUTER has the clear screen routine in absolute addresses (not changed by the VDMEM equate) so watch out if you move it from OCC00h!

I would be more than happy to share what I have worked out with anyone interested, and to hear from anyone who has done more! I do hope someone can help out with my 8085/VDM problem.

Sincerely

Bill

William T. Hole
260 Collingwood
San Francisco, California
94114

...ON SYSTEM NOISE PROBLEMS

Dear Mr. Vim Toutenhoofd,

I am sorry to hear that you are having such bad problems with your disk system. Make sure that you have a good heavy ground strap going from the metal chassis of the SOI to the metal chassis of the disk drive: the ribbon cable has too high an impedance to high frequency noise, and a solid ground is an absolute necessity! I use the braid from coax cable for my grounding straps. In addition, all peripherals such as printers are also grounded to the SOI via a good ground strap. You would be surprised how many subtle problems disappear when adequate grounding is used. I then have the SOI grounded to power system through a braided cable also, since the three prong outlets do not always give a solid ground, and ANY noise can wreak havoc in a computer system.

Dynamic memory CAN be a problem, unless it was specifically designed to be compatible with disk drives. The PROCESSOR TECH boards are a bit flaky in this regard. I am currently using a 48 K dynamic memory board made by MEASUREMENT SYSTEMS AND CONTROLS INC. This board has given me no problems at all. I have two of them, one running in my SOI, and the other running in a Z80 machine I designed, that thinks it is a SOL. I highly recommend the board. It is a clean design, WELL documented, easy to use, and quite reliable. They can be purchased in various memory sizes up to 64K. The 48K version is sold by MINI MICRO MART of SYRACUSE, NY for $589 assembled, tested, and with a 1 year guarantee. (The 48K board can be expanded to 64K by adding 4116 memory chips, which are about the least expensive 16K chips around). A nice feature of this board is that you can deselect portions of memory to allow ROM to exist within the address space normally occupied by the RAM.

I do not know the intimate details of the THINKERTOYS controller board, so I cannot guarantee compatibility, but it is a fact that the boards work flawlessly in both of my systems with the

MICROMATION controller that utilizes the READY lines for synchronizing data transfers. It also worked with my VISTA controller for 5" diskettes.

I wish you the best of luck in getting all your problems squared away, and hope that when you finaly get it all running, that you find this software useful.

Sincerely yours,

Fr. Thomas McGahee

Fr. Thomas McGahee S.D.B.

FR. THOMAS MCGAHEE 202 UNION AVE., PATERSON, NJ 07502

PS. Try using coax braid for your grounds: this may fix your memory problem as well!! The bigger the ground strap the better.

...ON VISTA DISK FOR SOL

Jordan L. Torgerson
22410 Barbacoa Dr.,
Saugus, CA. 91350
17 Sept 1980

Last year, I had volunteered to do a column on hardware compatability with the Sol. After the initial compilation of data was published, I ran dry - never got any more reports from Proteus members. That's what happened to that. Early this year, I changed careers, got married, and moved back to California at long last .. the dust is just now beginning to settle! When things settle down a bit more, I'd be willing to have another stab at the hardware column.

The next thing I will submit will be a hardware review on the Vista V-200 drive with it's CP/M and Basic-E compiler. I wanted to buy a Helios naturally, but alas, I can't afford one. The next, most logical choice would have been a North Star - but I also wanted CP/M and a compiler. For the same price as the North Star disk, Vista offered a system which included CP/M and Basic-E.

I called Vista in Santa Ana to find out about compatibility - you might be interested in the following:

They have modified the controller board, so that no modification of the Sol is necessary, as reported by Fr. McGahee in last years Kilobaud article.

They claim the system comes with the appropriate software and hardware to be able to literally plug it in and be up and runnigg.

The sales manager (or some sort of management person) said that they had a contract with the State of California to support 150 Sol's in the state library system, and he said that when PT went out of business, Vista bought up as many parts as they could from PT. He said that if my Sol ever needed work, Vista could do it, and that there was no question that the Vista V-200 would work in a Sol.

I inquired about memory access requirements - was worried about 16K Dynabyte board, which claims to be DMA-compatable, but has 300 ns chips. The tech at Vista said not to worry, that the V-200 DIDN'T use DMA, and that any memory board would work as long as the Sol was happy with it. That was good to hear!

The PROM Bootstrap on the controller board is addressed at 0000, but Vista said they would relocate it anywhere you needed it. I got the impression that they'd provide the relocated Prom at no extra cost as long as they were told about your requirement at the time of order, and, all Lifeboat sofware is available for the Vista system.

My Vista should arrive this week or next, and I'll write up a full report for Proteus when I get it running. (Will put it in the correct format too!) I have talked to a couple of TRS-80 Vista users and have had good reports - if the Vista works out, we may be able to negotiate a club discount or something. If I like the thing, and get good support from Vista, I'll be happy to approach them on this, with your approval of course.

Question - is there a local Sol group in the LA area? For that matter, is there any possibility of obtaining the names of LA Sol users to form a group if there isn't one?

Enough for now - thanks again for the fine work you have been doing on behalf of Sol-owners.

...ON FOR-NEXT LOOP FIX FOR EC BASIC

I just got around to implementing the fix for the BASIC REM statement described by Bob Werner in PROTEUS NEWS, vol. 3, no.1. In the description of the procedure Bob refers to an earlier fix for a FOR-NEXT problem that was published in ACCESS vol.2, no.1. I don't have access to that ACCESS, but fortunately I found that an update that I received with my BASIC dealt with the same problem. I think it would be useful to your readers to pass on that fix, especially as it specifically mentions that the information in ACCESS is incorrect.

The following is the gist of the fix for FOR-NEXT loops described in Processor Technology Extended Cassette BASIC Update 731064, page 7. The update has no copyright notice:

```
Load BASIC into memory using GET.
ENTR the following at 0B50: C1 CA 40 0B
ENTR the following at 3F81: FE 9D
SAVE BASIC 0 3F84
```

Yours sincerely,

Al

Albert S. Woodhull
33 Enfield Road
RFD 2
Amherst, Massachusetts 01002
1 August, 1980

...ON MCGAHEE, EXPANSION JACK J4, AND A SOUND GENERATOR

Dear Stan,

I have aquired quite a lot of information from Proteus News, and it's about time I add my two cents.
My SOL has SD Sales Expandoram board and I have had no problems with it after the modifications I read in the news letter. The mod's made to Ptc's ECBasic and ALS-8 are greatly appreciated. Fr. McGahee's Modifier II is a great help for someone who doesn't like the location of a program or just wants to change a few things, I recommend it highly. Everyone in the news letter has done a great job in helping the rest of us with the little problems that occur in programing, or just little things that make it work better. Thanks everyone and don't stop, some of us need lots of help. What about assembly language programming? Is their someone out there who can help us understand it all? Has anyone figured out what the display expansion jack (J4) is for or what we can use it for?

Oh no not again. I have a habit of getting off track sometimes. Well it's hard not to, I've gotten so much out of the news letter. Getting back to my two cents, I have recently purchased Digital Research Computers(of Texas) S-100 sound computer board along with their SCL software, and have found it to be a great sound board. It allows you,under complete computer control, to generate an infinate number of special sounds and effects for games or any program. I believe this board, with the right program, can produce synthesized voice. I have already produced several almost human sounding words, but they need ironed out a little. The board uses two AY3-8910 sound chips. It's amazing what this board can do. If anyone wants a good sound board to add realistic sounds to their games and programs, for under $85, here it is and it works well with SOL.
The SCL software,in eprom, needs an I/O routine
Here is the one I wrote:

```
START   LXI     H,INP       *FETCH A CHAR.
        LXI     D,OUTP      *PRINT A CHAR.
        CALL    SCL         *SCL EQU EOOOH
        CALL    SOL         *SOL EQU COO4H
INP     IN      KS          *KBD STATUS  EQU OFAH
        ANI     01          *CK DATA AVAIL.
        JNZ     INP         *LOOP
        IN      DATA        *DATA EQU OFCH
        ANI     07FH        *STRIP PARITY
        RET
OUTP    MOV     B,A         *READY DATA FOR OUTPUT
        CALL    SOUT        *SOUT EQU OC019H
        RET
        END
```

PS Anyone out there working with this sound chip drop me a line or two,maybe we can work on it together?

Donald F. Siebenrock
R.D.# 2
Clearfield, Pa. 16830

...ON Micropolis Users Group
604 Springwood Circle
Huntsville AL 35803

The Micropolis Users Group (MUG) is a newly formed organization serving all users of the Micropolis supplied DOS and BASIC. We publish monthly newsletters for a membership fee of $12 a year. One of our goals is to locate the existing software, and to encourage the development of new software, which will allow the Micropolis owner to perform all his computing needs without requiring CP/M or some other operating system.

...ON HYTYPE, ED BASIC, SAVE

SDS SOMERSET DATA SYSTEMS, INC.
P.O. BOX 436, 30 MORRISTOWN ROAD, BERNARDSVILLE, NEW JERSEY 07924
(201) 221-0966

October 7, 1980

Dr. Stanley M. Sokolow
1690 Woodside Road, Suite 219
Redwood City, CA 94061

Dear Stan:

Louis Bucklin's problem reported in the May/June PROTEUS News, that a Diablo drops the first character, is not restricted to the Sol-HyType interface. Mine does it too, but only with mylar ribbons. The direct cause is that the ribbon catches briefly on the daisy wheel when rising into position, and the underlying cause is that the daisy wheel has worked loose on its mounting. A firm push to reseat the daisy wheel will cure the problem unless the wheel is badly worn.

Some tidbits for novices:

EDBASIC (PTDOS 1.5 version) contains a function not listed in the manual, SYST(0). It returns the error code of the last system (PTDOS) error, and is presumably intended to be used when ERRSET has trapped the error and the ERR(0) function, which will always return "FS ERROR" for a system error, provides inadequate information about the nature of the error. The (0) is required by the syntax, although as with many if not all functions any expression that evaluates to zero may be used within the parentheses.

The bytes that make your BASIC semicompiled program files IAN-protected and other BASIC files I-protected are at addresses 1E4E and 1E41, respectively. They may be changed by POKEs for a specific occasion, or permanently by changing them and reIMAGEing BASIC. Section 5.2.3 of the PTDOS manual says which bits to set for what; the value to POKE is the decimal value of the binary number thus formed. If these locations don't contain 38H and 08 in your version, you may be able to find the correct bytes by finding the sequence CD BC BC 00 (Call SYS--CREOP) in the code and then backing up. This sequence starts at 1E55 in my version, and appears in only one other place, at 0B91 where it directs the creation of a file to be used for output if the one named after SET OF= does not exist.

To set the file-buffering option for dynamic buffering (apparently no one has ever felt the need to do so) the command in BASIC is SET FB=nonzero expression, not SET BF ditto.

And for those who don't care about BASIC:

To add a file to an archive without overwriting the existing contents, SAVE it in a new archive file first, then add the new archive to the old using COPY newarchivefilename, oldarchivefilename, S=A.

Sincerely yours,

Jay Parsons

...ON GREENLAW'S TAPEDISK FOR CP/M

Don Bosco Technical High School

202 UNION AVENUE, PATERSON, NEW JERSEY 07502
Telephone: (201) 278-8800

April 1Ø, 198Ø

Dear Proteus People,

See the attached announcement for more software for SOLs. This time it is a USER AREA for SOLs running with NORTHSTAR disks. I have lately written various custom USER areas for a number of SOL users with NORTHSTAR systems. Unfortunately, completely custom software takes lots of time and therefore costs a reasonable sum. I have taken the best and most useful features of the various routines I have written, and combined them with a number of different printer packages to create a sort of semi-custom set of USER area routines. I will create semi-custom routines for printers not specifically covered for only $15, and sell the existing routines for $1Ø. My record in shipping software is quite good. The longest delay was 4 days, due to a postal holiday. (That does not count truly custom software... that has sometimes taken as long as a whole week!)

The MODIFIER II has been quite well received. Did you know that the current version will run with ANY 8Ø8Ø or Z8Ø computer having memory-mapped video? (It prefers 16x64 and is very fond of SOLs, but gets along well with almost anything, regardless of screen format or screen address). Most of my stuff starts out as specifically SOL-2Ø software, but some of it is just too good to tie down to just one machine. (All my software DOES, however, come initialized for the SOL-2Ø).

Lewis Moseley sent me a copy of Richard Greenlaw's TAPEDISK routines, including MFDT. They are great! All my software destined for CP/M disk systems and supplied on tape will be supplied in TAPEDISK format henceforth. (All orders include at NO additional cost the necessary TAPEDISK routine to load the entire tape onto disk... Richard Greenlaw has allowed this, and is to be applauded for his approach to making his routines a true standard medium of exchange among SOL/CPM users).

Write to me and tell me what sort of utility software stuff you would like to see for the SOL. Who knows, if I find the time, I might just sit down and work on making some of it a reality. I am also willing to write various kinds of custom software for those who do not have the time or inclination to do such stuff on their own. My rates are reasonable, especially if the software is such as to be useful to a large number of people. (That's how I got involved in writing software for the NORTHSTAR. A few individuals asked for custom software, and from there I made up semi-custom packages that I can sell dirt cheap). If you do write, please include a self-addressed, stamped envelope if you request a reply.

Sincerely yours,
Fr. Thomas McGahee
Fr. Thomas McGahee S.D.B.
phone (2Ø1) 595-88ØØ

NORTHSTAR/SOL SOFTWARE FOR SALE

USER AREA FOR DOS: All versions listed include the following features: Run on SOL-2Ø with any version of NORTHSTAR DOS. Can be patched to run with CUTER. Mode select key acts like control/c. Backspace works on both screen and printer. Cursor control keys converted to NORTHSTAR codes. Delete converted to control/Z. Listings can be paused using space bar. Control/P causes BOTH video and printer to be active. Control/V activates video and deactivates printer. Control/P and Control/V are available through both the CIN and CONTC routines, and so are always accessible! Output devices may be specified via "A" register as follows: Ø=VIDEO 1=SERIAL 2=PARALLEL 3=VIDEO+PRINTER. Note that Control/P overides the above and forces BOTH VIDEO and PRINTER to be active. A special PAGING feature causes a FORMFEED to be sent after a user-defined number of line-feeds. Instructions tell how to deactivate paging feature, and how to cause a pause instead of a FORMFEED. (Allows use of sheet paper).

Order software by printer type:
NSTARTTY is for TTY and similar printers connected to serial port #1. No handshaking, but allows delay after CR and LF. Use this with TTY, TERMINET, SELECTRIC, or any other terminal that does not have overflow problems.

NSTARSER is for serial printers on port #1 that tend to overflow their buffers. This version allows a slight delay to be generated after each character, with longer delays after CR and LF (and internally generated FORMFEEDS). Use this for printers without handshaking.

NSTARH14 is for HEATH MODEL H14 serial printers attached to serial port #1. Provides handshaking via Control/Q and Control/S. Allows operation at ANY baud rate without overflow.

NSTARDBL is for DIABLO 161Ø/162Ø and compatible printers. By compatible I mean any serial printer on port #1 that allows handshaking using the ETX (Ø3) and ACK (Ø6) codes. Allows operation at any baud rate without overflow. Assumes the printer has a buffer. (Is set for 16Ø character buffer, but buffer size is easily changed in software).

For other printers, including parallel types, write describing your printer's characteristics. I will be happy to customize for any printer. Please send self-addressed, stamped envelope with enquiry.

PRICES: Any printer package I have in stock costs $1Ø. Customized versions cost $15. Add $5 if outside continental USA. Price includes instructions, hardcopy listing on sheet paper, and program on inexpensive cassette in SOLOS/CUTER format at 12ØØ baud. Tape is in two parts: USER AREA and PATCH for DOS JMP TABLE. Orders normally shipped within 24 hrs. Send order to:

Fr. Thomas McGahee PHONE (2Ø1) 595-88ØØ
Don Bosco Tech
2Ø2 Union Ave.
Paterson, NJ Ø75Ø2

*** IF DOS ORIGIN NOT 2ØØØ, PLEASE GIVE ORIGIN ***

June 6, 1980

Northstar Users Association
131 Highland Ave.
Vacaville, CA 95688

Dear N* Persons,

I read the announcement of the formation of the N* users' group in PROTEUS, and am writing to inform you of some software that I have written that is of particular use to SOL users with N* disk drives.

As you know, the USER area is the software section of N* DOS that handles all the I/O requirements such as keyboard, printer, etc. I have written a series of custom USER areas for the SOL that support a variety of printers. You will notice that it is possible with these new USER areas to toggle the printer on and off from the keyboard as well as under program control. The VDM can be made to echo what is sent to the printer... this is particularly useful when the printer is separated from the computer by a fair distance. Listings can be PAUSED at any time, and you can even switch from VIDEO to VIDEO+PRINTER (or vice-versa), while a listing is taking place!

I am including a partial copy of the user notes so you can get a better idea of what the programs do. If you have any questions, drop me a line or call me at (201/ 595-8800). I would appreciate it if you would include the enclosed advertisement in your newsletter. By the way, I am available to do custom software at reasonable rates.

Thank-you for your consideration, and I hope that some of your readers will find my programs a solution to their problems.

Sincerely yours,

Fr. Thomas McGahee

Fr. Thomas McGahee S.D.B.

PS: Software comes on tape in SOLOS/CUTER 1200 baud format.

CUSTOM USER AREA FOR SOL/NORTHSTAR * BY FR. THOMAS MCGAHEE

INCLUDES PAGING FEATURE

Written for Sol-20 Users by
Fr. Thomas McGahee
202 Union Ave.
Paterson, NJ 07502

LOADING THE ROUTINES FROM TAPE

The I/O routines are supplied on tape in SOLOS/CUTER format at 1200 baud. The procedure is as follows: (material that has been underlined indicates commands typed into the computer. (CR) indicates a terminating carriage-return).

1) Turn on SOL computer.

2) Turn on disk drive, and insert diskette into Northstar drive.

3) Type EX E900 (CR) to boot up Northstar DOS.

4) Insert a diskette with factory-supplied DOS into drive.

5) Type LF DOS 5000 (CR). This will load a copy of the Disk Operating System at 5000H.

6) Reset SOL using REPEAT/UPPER-CASE.

7) Insert tape into cassette player. It should be positioned just after the end of the leader.

8) Type GET USER 5900 (CR). The computer should load the program USER starting at 5900H. (It is ORIGINED for 2900H, but we are loading it into the DOS image we have in memory at 5000H).

9) Type GET JMPS 500D(CR). The computer should load the program JMPS starting at 500DH. (Again, this is into the DOS image at 5000).

***** At this point, the new USER I/O routines have been loaded, and the Northstar JUMP table has been patched into the DOS image at 5000H. Now Northstar must be reentered and the new DOS with its I/O routines saved on disk.

10) *** REMOVE FACTORY DISKETTE *** and put another diskette in drive.

11) Type EX 2028 (CR). This will reenter Northstar.

12) Type SF DOS 5000 (CR). This will save the DOS at 5000H onto the diskette.

13) Reset SOL using REPEAT-UPPERCASE.

14) Boot up Northstar by typing EX E900 (CR).

***** At this point you should have your NORTHSTAR up and running with the new I/O routines. If you experience a problem, it is either because you did not follow the loading instructions exactly, or you do not have the printer baud rate and the SOL baud rate set up for the same value. See APPENDIX for information concerning program options.

Assuming that step #14 resulted in a successful boot, you can now check out some of the new I/O facilities.

USING THE NEW I/O ROUTINES

The following keys have been handled in a special manner.

CONTROL/C This key can be used to terminate listings, interrupt BASIC programs, etc. It functions exactly as outlined in the NORTHSTAR manual.

MODE-SELECT This key is the same as CONTROL/C. In most cases it is more convenient to use MODE-SELECT because of its size and position.

CONTROL/] This allows a NEW PAGE to be forced at the printer. It can be given ONLY at the keyboard.

BACKSPACE (CONTROL/H) is used by some software vendors to effect a backspace. This is translated so that on the video screen it reacts the same as a LEFT-ARROW, but it is sent to the printer as the BACKSPACE character. Some printers may simply ignore this.

MCGAHEE CONTINUED FROM PAGE 35

LEFT-ARROW This is changed to BACKSPACF, since the SOL does not have a backspace key.

RIGHT-ARROW This is also changed to BACKSPACE.

DELETE This is changed to UNDERLINE (It DELETES internally, and backspaces on video, but prints as underline on printer).

CONTROL/P This key is not echoed. It will cause all further output to be sent to BOTH the VIDEO and the PRINTER. It is possible to change-over in the middle of a listing, since CONTROL/P is decoded during regular character entry, and also during CONTC. If a CONTROL/P is sent during a listing, it will activate VIDEO+PRINTER, but print activity will not resume until some other key is hit. So CONTROL/P can be used to pause a listing that you are printing.

CONTROL/V This key is not echoed. It will cause all further output to be directed to the VIDEO only. It is the opposite of CONTROL/P. It may be used AT ANY TIME to switch from CONTROL/P VIDEO+PRINTER mode to simple VIDEO ONLY mode. If CONTROL/V is sent during a listing, it will activate the VIDEO ONLY mode, but screen activity will not resume until some other key is hit. Thus CONTROL/V may be used to pause a listing on the VIDEO device.

LOAD Is converted to a COMMA. This gives you a comma key next to the numeric keypad.

CONTROL/L (FORMFEED). When sent to the PRINTER, it resets the page counter and forces a new page. (Not all printers support FORMFEED, so on some this fuction is synthesized using calculated line feeds).

The special printer routines are accessed via a custom output routine. This means that the following output device codes exist:

#Ø **The VIDEO DISPLAY.**
#1 **The SERIAL PRINTER WITH PAGING.**
#2 **The PARALLEL PORT.**
#3 **Both the VIDEO and PRINTER (WITH PAGING).**

In addition to accessing output devices by using their pseudoport numbers, CONTROL/P and CONTROL/V allow the current pseudoport to be set from the keyboard. The user may ALSO set the current pseudoport number by stuffing the number into memory at ØC8Ø7H. In BASIC this may be done via an instruction such as "FILL 512Ø7, Ø3". DO NOT ATTEMPT TO USE THE SOLOS "SET" COMMAND TO SET THE PSEUDOPORT # PRIOR TO BOOTING UP NORTHSTAR, AS CONTROL MAY TERMINATE WITHIN SOLOS INSTEAD OF "USER".

PAGING FEATURE

The printer will perform an automatic FORMFEED when a preset number of line-feeds has been sent. The number is currently set at 5Ø. Some printer terminals ignore FORMFEEDS, and so for them we synthesize a FORMFEED by using calculated line feeds.

To change the number of lines to a page, simply change the byte at 29FF to the HEXADECIMAL value desired.

To force a new page, simply send a FORMFEED code to the PRINTER ROUTINE.

As an added convenience, a CONTROL/] from the keyboard will immediately reset the page counter and force a FORMFEED opration at the printer.

See APPENDIX for sheet-feed option. When using sheet feed option there is a pause at the end of each page until ANY key is hit. It is for THIS reason that we use CONTROL/] to signal a forced NEW PAGE, rather than the FORMFEED character. THE CHARACTER IMMEDIATELY FOLLOWING A CONTROL/] WILL BE USED TO REACTIVATE PRINTING FOLLOWING THE NEW FORCED PAGE. THIS MEANS THAT THE CHARACTER WILL BE "LOST".

To PERMANENTLY change the number of lines per page, do the following:

1) Under NORTHSTAR, type LF DOS 5ØØØ (CR). This will load a memory image of the DOS at 5ØØØ.

2) RESET the SOL using REPEAT/UPPER CASE.

3) Type EN 59FF (CR). This allows changing the byte at 59FF, which is the memory image of the DOS location 29FF.

4) Type the HEXADECIMAL value of the number of lines you want per page, followed by /(CR), to end the entry mode.

5) Type EX 2Ø28 (CR), to re-enter NORTHSTAR.

6) Type SF DOS 5ØØØ (CR). This will save the DOS image on disk.

From now on, whenever the disk is booted up, the new page length will be active.

See APPENDIX for details on deactivating PAGE function, and changes to be made if using sheet paper.

APPENDIX FOR NSTARNUL NORTHSTAR/SOL/TTY-LIKE PRINTER

NSTARNUL IS FOR SERIAL TERMINALS THAT DO NOT PROVIDE HANDSHAKING. To allow them to operate properly, LF is given a delay after it is sent. FORMFEED is accomplished by sending enough linefeeds to bring up the next page.

Should you ever want to DEACTIVATE the paging feature, simply replace the LDA COUNT at 29AB with a RET instruction. This replacement can be performed on a temporary basis using POKE instructions in BASIC, or using your MONITOR. Permanent changes can be done by altering the DOS image (as we outlined above when we told you how to alter the page length).

IF USING SHEET PAPER, change the JMP LFDLY at 29C3 to a CALL CIN. This will cause a wait for any key at end-of-page.

There are 29 EMPTY locations starting at 29E1. These may be used to extend the CONVERSION TABLE that begins at 29D6. To extend the CONVERSION TABLE, start new entries at 29EØ, and place ØFFH after last entry. Remember that the most significant bit is stripped only AFTER conversions have been performed, so that the special SOL keys can be detected.

MCGAHEE CONTINUED FROM PAGE 36

NORTHSTAR DOS RELEASE 5.1S has a bug, in that when giving the message "PRESS RETURN TO CONTINUE", it is missing a CR/LF. It is helpful to replace the "TO CONTINUE" portion with spaces and a terminating ØD, ØA (CR/LF).

BASIC programs may access the printer via "FILL 512Ø7, Ø1", and deactivate the printer via "FILL 512Ø7, ØØ". BOTH the VIDEO and PRINTER may be activated by using "FILL 512Ø7, Ø3". DO NOT use SOLOS "SET" commands before booting up NORTHSTAR. Use CONTROL/P and CONTROL/V to switch the printer on and off, or make sure that the proper value is stuffed into ØC8Ø7H (512Ø7) via BASIC.

My thanks to Dr. Carl Rothe for his suggestions for improving upon my original program and documentation.

ADVERTISEMENTS

ADVERTISEMENT: I have custom USER routines for NORTHSTAR that include drivers for a wide variety of printers operating under control of a SOL-2Ø. Each includes the following features:

FULL INSTRUCTIONS AND ASSEMBLY LISTING INCLUDED.
MODE SELECT is converted to Control/C.
LEFT-ARROW is converted to BACKSPACE
RIGHT-ARROW is converted to BACKSPACE
BACKSPACE works on both VIDEO SCREEN and PRINTER.
LOAD is converted to a COMMA for numeric keypad use.
USER MAY EASILY ADD HIS OWN CONVERSIONS!
CONTROL/P can be used to cause PRINTER to ECHO VIDEO!
CONTROL/V can be used to return to VIDEO-ONLY mode.
CONTROL/V/P ARE AVAILABLE AT ALL TIMES, EVEN DURING LISTINGS!
OUTPUT DEVICE may be selected as device #Ø-3.
DEVICE #3 is defined as BOTH VIDEO and PRINTER.
PAGING on all PRINTER output is automatic.
PAGING can be patched to support single-sheet paper.
FORMFEED works even with TTY-type terminals.
PRINTING MAY BE PAUSED.

PRINTERS SUPPORTED

SERIAL TELETYPE WITH NULLS AFTER LINEFEED

NSTARNUL Connects via SOL Serial Port #1. Provides delay following a linefeed by sending out a user-defined number of nulls. Comes set for 1Ø nulls. FORMFEED is accomplished with linefeeds. MAY BE USED WITH ANY SERIAL PRINTER.

SERIAL WITH HANDSHAKING VIA CLEAR-TO-SEND

NSTARCTS Uses SOL Serial Port #1, and samples CTS via pin 5 of J1. This allows operation with any device that communicates ready status using this signal. (At PRINTER this is called RTS).

SLOWED SERIAL via SOL Serial Port #1.

NSTARSER has no handshaking, but it does allow the user to specify a small delay following EVERY character, and longer delays after LF. Use this version for serial printers that tend to overflow their buffers. The small delay can be "tweaked" by the user to allow serial printers to run at high baud rates without loss of data. FORMFEED is accomplished with linefeeds.

SERIAL HEATH H14 via SOL Serial Port #1.

NSTARH14 INCLUDES HANDSHAKING using Control/Q and Control/S. Allows operation at ANY baud rate without loss of characters.

SERIAL DIABLO 162Ø (or equivalents such as IPSI 1622)

NSTARDBL Connects via SOL Serial Port #1. Allows use at ANY baud rate without loss of data. Uses ETX (Ø3) and ACK (Ø6) control codes for handshaking, so will work with ANY serial terminal that uses ETX and ACK codes.

OTHER TERMINALS including PARALLEL types:

Send complete information on your printer needs including a stamped self-addressed envelope, and I will quote a price for customizing a USER routine specifically for your terminal. New routines are constantly being added, so chances are good that I will already have one to meet your needs.

I also provide other customization services. If you have a specific need, write and let me know. Please include a stamped self-addressed envelope with all enquiries. If you belong to a computer club, or know of other SOL users who might benefit from any of my software, I would appreciate it if you would let them know about what we have to offer. I operate on a shoestring budget here, and rely upon announcements in club newsletters, and word of mouth to provide advertising.

Currently stocked USER routines sell for $1Ø in continental USA, $15 elsewhere, including CANADA. Routines written for printers not currently stocked cost $15 in continental USA, $2Ø elsewhere.

Send check or money order to:

Fr Thomas McGahee
2Ø2 Union Ave.
Paterson, New Jersey Ø75Ø2

Other programs available (These run on ANY 8Ø8Ø/Z8Ø).

MODIFIER II: utilities for 8Ø8Ø/Z8Ø and memory-mapped video. For folks who like to dig around inside programs. Includes an easy to use relocator. View memory on screen. Find/Dump/Enter strings to memory. Find/Dump/Enter hex. Display pages of memory on screen. And more.

SOLO5: an on-line mini screen editor for 8Ø8Ø/Z8Ø and CP/M with memory-mapped video. Perfect for use with ED and BASIC. Allows insertion and deletion of characters, visual expansion of tabs, and much more. Allows BASIC lines to be "moved" without retyping the whole line! For people who like to use their computer without having to learn the editing peculiarities of every new program they run.

Send SASE for full details.

BUG IN GERRY FRICKE'S FOOTBALL

You might note that there is a minor bug in Gerry Fricke's FOOTBALL game on disk H1. Line 1160 says IF . . THEN: GOTO 1190 when it should put the GOTO 1190 on the next line so that it operates regardless of the truth of the condition. As it stands one player's field goal attempts go backwards some of the time.

Keep up the good work,

Jay Parsons

...ON A DISK MODIFICATION

Don Bosco Technical High School

202 UNION AVENUE, PATERSON, NEW JERSEY 07502
Telephone: (201) 278-8800

December 10, 1979

Dear Mr. Vim Toutenhoofd,

My disk modification involves installing an optically-isolated zero-crossing solid state relay in line with one of the motor drive wires. The relay gets power from the head load relay. The violet wire is + (positive). I connect the two wires I soldered to the head load relay (actually it is a solenoid) onto the solid state relay. There is plenty of room inside the Thinker toys drive cabinet for mounting the relay. Now, whenever the head loads, the motor is on. During idle times, the motor is off. It turns on very smothly, with NO transient noise. I have had absolutely no trouble with this, and I have three drives using this modification. The Solid state relay is stock # 519-ECC-D1Z1C cost $12 from Mouser Electronics, 11511 Woodside Ave, Lakeside, California 92040 orders less than $20 require a $3.00 handling charge, + postage for 1 pound.
(This is an EXCELLENT SOLID STATE RELAY!)
The modification should work on your system with no problems. No charge for the "CONSULTATION" - just let others know about it if you find it useful.

Sincerely yours,

Fr. Thomas McGahu, SDB.

......ON SCELBI'S "PIMS"

I have been attempting to implement Scelbi's "PIMS" program on my computer, but have so far been unsuccessful. The problem I have encountered is in the cassette handling subroutines. How does one pass strings and data to and from cassette during a program? I have tried using the DEFUSR and USR statements to call the CUTER tape subroutines FOPEN, FCLOSE, WRBYT, RDBYT, WRBLK, and RDBLK, but this does not seem to work properly. Can any reader of PROTEUS suggest a solution to this problem. N.B. I am using the G2 Microsoft Extended Basic version 4.7 rather than PTBasic as the latter was not available locally when I bought my CUTS and GPM boards.

Yours truly, Alastair S. Preston
15D Twin Terrace,
Edmonton, Alberta,
CANADA, T6K 1V4
23 July 1980

...ON A PROGRAM CORRECTION,ETC. November 29, 1979

Dear Stan,

Please print the following patch to program 3 on page 19 of Vol. 2 No. 4 :

```
CAF0 78              WRITE     MOV   A,B
CAF1 FE 0D                     CPI   0DH
CAF3 CA FD CA                  JZ    WRITE0
CAF6 FE 20                     CPI   20H
CAF8 D8                        RC
CAF9 AF                        XRA   A
CAFA C3 01 CB                  JMP   WRITE1
CAFD 3A 61 CB       WRITE0     LDA   FLAG
CB00 3C                        INR   A
CB01 32 61 CB       WRITE1     STA   FLAG
CB04 FE 02                     CPI   2
CB06 F0                        RP
CB07 00 00                     FILL 2,0
```

NOTE: SCRATCH the old version and ASSM new version

This patch restricts the byte mode cassette recording to a single <CR> byte between lines of BASIC program text thus making the recordings match the Cassette Software Library standard.

Some general comments ----

1. Does anyone know of a place I can get a SOLOS personality module which uses 2708 EPROM's ?
2. Moseley asked about being able to read TRS-80 cassettes. I too have thought about this (and APPLE cassettes) but have always given it up because of the differences in BASIC language syntax (especially the input/output to the display, i.e. high and low resolution graphics).
3. Does anyone know if a SOL-20/MICROPOLIS version of PASCAL is available or in the works ?

Keep up the good work with PROTEUS NEWS.

MELVIN M. DALTON
7826 WEST 80TH ST.
PLAYA DEL REY, CA 90291

...ON HORRORS'

MENCO SERVICES September 21, 1980
4877 Fairlawn Circle • P. O. Box 3511 • Boulder, Colorado 80307

Dear Stan:

First, with regards to the PROTEUS editor's response to the dentist in Nepal, I want to pass along my experience with my North Star drive. I operate in a room where the ambient temperature is 76 degrees. Frequently, while operating with no thought of accessing the drive, it will start running. On several of these occassions, I would discover later that the disk directory had been destroyed. Needless to say, I now have backup disks to contain the really important stuff.

The second thing is kind of a "horror" story. Back in May, John Dvorak announced the availability of a North Star Basic version of "Adventure" which I promptly ordered. The disk was received and I sat down to discover what "Adventure" was all about. I read the "READ-ME" file and followed the directions. Upon commanding LOAD ADVENT, if that was the name, I promptly got the TOO LARGE OR NO PROGRAM error message. (My SOL has a 48K RAM.) Repeated tries resulted in the same diagnostic. I calculated that I had received a faulty disk and sent it back to Dvorak along with a detailed explanation for doing so. I have tried to communicate with him on a half-a-dozen occassions (enclosing SASE or stamps a couple of times). To date, I have heard nothing from him and am about to conclude he thinks he needs my money a lot worse than I do. Moral: Pure frustration when dealing with "fly-by-nighters".

PTC SOURCE CODE NEWS

By the time you read this, all buyers of the PTC source code should have received a letter from Proteus with a license agreement and an order form for the media on which the programs are to be sent. (See the last issue of volume 2 for details.)

We expected to ship the programs by now, but our real work interfered with Proteus projects, and we encounted some unexpected obstacles. These are on the road to solution, and are explained in the letter.

In the next issue, we will re-announce the available source code programs so that other members can consider purchasing what they want.

CALL FOR MORE ARTICLES AND LETTERS

Our file of material to publish is running low due to the extra size of this double issue. Please, if you have been thinking about writing something for Proteus, take the time to do it now. Review a product you have used, tell us how you did some neat trick, ask a question, answer someone's question, whatever.

Please, when you do write, it helps us a great deal if you will simply buy a new ribbon for your printer and use it for the article you intend to send. You will need the new one anyway, so take the time to do it right. Faded ribbons make printouts that are not camera ready, or that reproduce with hard-to-read broken letters. If we have to retype a long article, we usually don't, unless it is exceptionally important for our readers.

If you don't have a good printer for this purpose, you can send us an article or letter on a cassette or Helios diskette. Cassettes recorded in byte-mode, as 256 byte blocks the way that Extended Cassette BASIC saves "text" form, or by the Helios driver "CTAPE", are the easiest for us to read. But ALS-8 compatible files are acceptable, too.

BASIC COMPUTER GROUP SOFTWARE UPDATES

If you have the BCG's software for Sol/Helios systems, please examine the list of latest revision numbers shown below. If your software isn't up to current levels, I will update it for a fee of $20 per disk, which includes copying onto your original disk (marked by the manufacturer), an update sheet (if needed) explaining the changes, and postage to you.

Latest version numbers of system disks in the Basic Computer Group Software for Helios/Sol systems:

Wordwizard 4.0.2
Mailmaster 3.0.2
Mailsort 1.0.1
Accpac General Ledger 1.3.4
Accpac Financial Report 1.1.2
Accpac Accounts Receivable-Daily 1.0.1
Accpac Accounts Receivable-Monthly 1.0.1
Accpac Accounts Payable-Daily 1.0.2
Accpac Accounts Payable-Monthly 1.0.3
Accpac Programmers Package 1.1.0

COMING ATTRACTIONS

In the next issue we will have an article by Joe Maguire on maintaining and repairing your Sol, with some things that anyone can do even if you have no great technical skills. He has also given us instructions on how to relocate your Sol to F000 address space, with the ability to switch back to C000 just by reinserting your old personality module. And Stephen Maguire, Joe's son and apparent programming wizard of the team, has given us an I/O routine for using the North Star Pascal Version 1.0 with the Sol. We also have an article from Royce Bacon on how Extended Cassette BASIC store source code in memory and a cross-reference program for ECBASIC, written in BASIC.

Jack Kinney has modified the Sol keyboard to provide automatic repeating of any key held down beyond a certain delay. There will also be news of more software for Sol. We'll also have a product review or two. Other goodies are yet to be revealed.

MEMBERSHIP LISTS AVAILABLE TO MEMBERS NOW

We have often received requests from members who would like to know the names and addresses of other Proteus members in their area. In the past it was too difficult for us to do this, but now we have the hardware and software to make it possible. So, for a service charge of $10, we will send you a list of members in the vicinity of your ZIP code. Foreign subscribers are not on this list, but their addresses are available separately, also for a $10 service charge.

Note that only a regional list will be sent. The full roster is not available for purchase. So, when you order, be sure to tell us your desired ZIP code. We will send approximately 120 names who fall closest to your ZIP code in our latest mailing list.

More Unclassified Ads

FOR SALE: SOL -20, 16K static RAM, 9" monitor, Voltage spike protector, TREK-80, EC BASIC, $1075 negotiable.
George Atwood 2119 Eunice St. Berkeley, CA 94709 415-525-3867

COMPU-COVER for SOL-20, and other computers, printers, monitors etc. Cloth backed vinyl. $14.95 plus 1.50 postage and handling. Write for full list of covers available. Send dimensions for price quote on non-standard covers.
Mary Esther P.O. Box 324, FL 32569 (904) 243-5793.

WANTED TO BUY: 4-Slot HELIOS II.
Allan Olson, 474 Ridge Road, Nevada City, CA 95959

TABLE OF CONTENTS

SUBSCRIPTION RENEWAL FORM

Please use a photocopy of this form, or just send us the requested information, along with a check in U.S. funds for your subscription renewal. $24 per year for U.S.A., Canada, and Mexico. $32 per year for other foreign addresses.

Name:

Mailing Address:

City, State, Zip:

Country, if foreign:

We will make your name and address available to other members on a roster, unless you check this box.

[ ] Please keep my name private.

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, USA.

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

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