EDITORIAL:

WHEN I'm 3:00 A.M. in the morning and I've just completed the final pass of my 1st issue of PROTEUS, all that remains is my first editorial. It's amazing to think of all the changes that have occurred 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 plasma and simple fact that the Sol20/ Helios was the best of all the micro's and continues to be the best, in my opinion. Now Word Wizard has yet to be duplicated by any other micro. As a programmer aid, PT DOS 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 lowest 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 of PROTEUS has been going. 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 already are not already. As Processor Tech. equipment owners need to support each other as much as possible. One way of doing this is to find all the P.T.'s owners out there, the 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 shops 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 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 occurred to me (2 do my next thinking at 3:30 A.M.) that many schools (High Schools to be specific) bought Sol's and then, when the computer hobbyist/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 Merrier

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 form. I began putting an innocuous character following REM. While this procedure 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 Processor Tech. list 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 skips over any blanks and returns the first character it finds. Then the loop compares the characters. If they match, routine 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 subroutine 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 'skip 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 'skip 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.
2. Do not XEO BASIC--you want to simply load it into memory at this time.
3. Use the Solos JUMP command to Print the contents of the affected area.

   BUMP ED EF

Verify the contents of memory against the 'old' listings 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' listings.

   Continued on p. 2.
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 to the error. You need to change the checksum as well. So, again, use the DUMP command to display the recorded checksum.

   DUMP 0FB1 3FB6
   This should display the values FE 9B. Assuming you have already
   incorporated Processor Technology's fix for a FOR-NEXT Problem
   that was published in ACCES Vol. 2, No. 1, again: do not
   proceed unless the values match. To avail yourself of this
   service, prepare a clean copy of the program on tape in CUS 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 $5 for each EPROM
   that must be erased. For my part, I will verify each blasted EPROM
   against the tape copy. In case of programs that require multiple
   EPROMs, I will affix 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 PYS, 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: CUS format
   tapes at 1200 baud; 8" CP/M compatible disk in either single or
double density SOFT SECTOR; 5 1/4" SOFT SECTOR CP/M compatible
diskettes (disk has a single sector hole; 2708 EPROM; 2716 EPROM
(5 volt type). I also hope within the near future to support 5 1/4"
double density soft sector, but do not at present. There is a $5 fee
for such media transfers, with you supplying all pertinent medi,
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 RAMS and
   PLYTRPs should be the program be assigned?] If at all possible,
   send media that contain UNL's the programs 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 transferring to and from tape, the $5 charge applies to each 2K of data that is transferred.

A rather extensive set of 6800 utilities have been written by myself and bro. Al Roman. They will be offered on C8TS tape and on disk (in the disk format 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 SOLUS, but is easily modified for any MOS computer. (Its sole interface to the basic 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 PROMUS. A brief description of its utilities is included here for those who 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 sessions of memory with any code. It moves 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 42 characters in length. Note that the HEX and ASCII searches result in the found code being shown on the screen in CONFOR, 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 pager 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 relocating similar to that which SOLUS and Mouse uses for relocating the 6800 is also included. This relocator allows simple entry of all necessary parameters. In 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 as long as you know where the data areas are located.

The above-mentioned utilities bear the name THE MODIFIER II. It is offered on C8TS tape for $10, on C8TS disk for $15, including documentation. As a service to interested users, an assembly listing is also available. It is $10 on C8TS cassette in Processor Tech Assembler format (written on the 6800/6800), 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 C8TS/Assembler format. Ordered separately, it is $13. If I put the program and assembler on the same disk, you get both for $25, including a hardcopy listing. WHEN ordering, please specify tape or disk, and if disk, whether in soft sector or 5 1/4" soft sector. (If you want double density 6", please add $2 per disk, as the double-density disk costs me more). Note: 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 efforts). Every effort will be made to provide fast service. Please help by including all pertinent information with your order.

Sincerely yours,

Fr. Thomas McGahan, S.O.L.

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

Dear PROMUS,

For the last year, I have been looking for the "Computer Dead Room" at BYTE Shop in San Rafael. I am now working out of my home, and I have all of the necessary tools to repair the SOL and align the Helios.

My rates are $25.00 per hour plus $1 per mile for a total rate of $40.00. I will pick up and deliver anywhere in the greater San Francisco Bay Area.

Thank you

Albert A. Bonelli III
P.O. Box 394
Petrolia, CA 95553
(707) 62-9514

NOTES

July floppy diskettes seem to not have a good polish in the recording area.

Georgia Magnetics have a good polish.

NKRA INFORMATION

1. PROMUS P.T. PART #
   U37 AND U31...........26107
   U22....................26108
   U30....................26110
   U56....................26111

2. THE NKRA DELAY LINE IS 00002...ONE SOURCE WAS:

ENGINEERED COMPONENTS CO. (Part # S7LDM-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 "ENO" with identical start and end addresses.

"ENO" creates a file of length zero as an end mark, but Solos recognizes no end-of-file mark. Your convention is your decision, but the file "ENO" is commonly used.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>2-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>4-5</td>
</tr>
<tr>
<td>AND</td>
<td>4-17</td>
</tr>
<tr>
<td>APPEND</td>
<td>4-16</td>
</tr>
<tr>
<td>Arithmetic operators</td>
<td>2-4, 4-14, 5-16, 7-1</td>
</tr>
<tr>
<td>Arrays</td>
<td>5-16, 7-1</td>
</tr>
<tr>
<td>Arrows</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>ASC</td>
<td>5-14</td>
</tr>
<tr>
<td>ASC codes</td>
<td>A4-1</td>
</tr>
<tr>
<td>ATN</td>
<td>5-6</td>
</tr>
<tr>
<td>BASIC</td>
<td>1-1</td>
</tr>
<tr>
<td>Blanks</td>
<td>2-4</td>
</tr>
<tr>
<td>BYE</td>
<td>2-2, 5-32</td>
</tr>
<tr>
<td>C (Comma format)</td>
<td>5-25</td>
</tr>
<tr>
<td>C (Semi-compiled)</td>
<td>5-11</td>
</tr>
<tr>
<td>Call (Function)</td>
<td>3-11</td>
</tr>
<tr>
<td>CALL (Machine routine)</td>
<td>6-3</td>
</tr>
<tr>
<td>Calculator mode</td>
<td>2-6</td>
</tr>
<tr>
<td>Checksum</td>
<td>2-38</td>
</tr>
<tr>
<td>CHR</td>
<td>5-14</td>
</tr>
<tr>
<td>CLEAR</td>
<td>3-9</td>
</tr>
<tr>
<td>CLOSE</td>
<td>5-22</td>
</tr>
<tr>
<td>Colon (:)</td>
<td>2-3</td>
</tr>
<tr>
<td>Comma</td>
<td>4-4, 5-27</td>
</tr>
<tr>
<td>Commands</td>
<td>2-1, 5-31, A1-1</td>
</tr>
<tr>
<td>CON</td>
<td>7-2</td>
</tr>
<tr>
<td>Constants</td>
<td>2-4</td>
</tr>
<tr>
<td>CONT</td>
<td>3-8, 4-9</td>
</tr>
<tr>
<td>Control characters</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>Control X (X)</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>COS</td>
<td>5-33</td>
</tr>
<tr>
<td>CR - Return</td>
<td>1-3</td>
</tr>
<tr>
<td>CORD</td>
<td>5-33</td>
</tr>
<tr>
<td>D (Format default)</td>
<td>5-25</td>
</tr>
<tr>
<td>DATA</td>
<td>4-6</td>
</tr>
<tr>
<td>DB (Display ASC)</td>
<td>5-32</td>
</tr>
<tr>
<td>DEF FN</td>
<td>5-6, 5-8</td>
</tr>
<tr>
<td>DEL (Command)</td>
<td>3-14</td>
</tr>
<tr>
<td>DEL Key</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>DIM</td>
<td>5-11, 5-17</td>
</tr>
<tr>
<td>Dollar Sign ($)</td>
<td>5-25</td>
</tr>
<tr>
<td>DS (Display speed)</td>
<td>6-5</td>
</tr>
<tr>
<td>EDIT</td>
<td>3-6</td>
</tr>
<tr>
<td>EJSE</td>
<td>4-20</td>
</tr>
<tr>
<td>EOF</td>
<td>5-23</td>
</tr>
<tr>
<td>END</td>
<td>4-9</td>
</tr>
<tr>
<td>ERR(0)</td>
<td>5-31</td>
</tr>
<tr>
<td>ESCRCA</td>
<td>5-30R</td>
</tr>
<tr>
<td>Error messages</td>
<td>A3-1A</td>
</tr>
<tr>
<td>ERRESP</td>
<td>5-30R</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>2-1, 2-3</td>
</tr>
<tr>
<td>EXIT</td>
<td>4-14</td>
</tr>
<tr>
<td>EXIT</td>
<td>5-5</td>
</tr>
<tr>
<td>Exponential format</td>
<td>5-26</td>
</tr>
<tr>
<td>Expressions</td>
<td>2-5</td>
</tr>
<tr>
<td>FILE</td>
<td>5-19R</td>
</tr>
<tr>
<td>Floating point format</td>
<td>5-26</td>
</tr>
<tr>
<td>FN(1)</td>
<td>5-6, 5-8</td>
</tr>
</tbody>
</table>

INDEX to Extended Cassette BASIC Manual

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>END</td>
<td>5-6</td>
</tr>
<tr>
<td>FOR</td>
<td>4-12</td>
</tr>
<tr>
<td>Formatted PRINT</td>
<td>5-24</td>
</tr>
<tr>
<td>FREE(0)</td>
<td>5-18</td>
</tr>
<tr>
<td>Free format</td>
<td>5-26</td>
</tr>
<tr>
<td>Functions</td>
<td>5-5, A2-1</td>
</tr>
<tr>
<td>Functions-user defined</td>
<td>5-6</td>
</tr>
<tr>
<td>GET</td>
<td>2-1, 3-14</td>
</tr>
<tr>
<td>GOSUB</td>
<td>5-2</td>
</tr>
<tr>
<td>GO TO</td>
<td>4-10</td>
</tr>
<tr>
<td>Hexadecimal to Decimal</td>
<td>A5-1</td>
</tr>
<tr>
<td>IDN (Identity matrix)</td>
<td>7-2</td>
</tr>
<tr>
<td>IF</td>
<td>4-18, 4-20</td>
</tr>
<tr>
<td>INPUT</td>
<td>4-3, 5-29</td>
</tr>
<tr>
<td>INT</td>
<td>5-5</td>
</tr>
<tr>
<td>Integer format</td>
<td>5-26</td>
</tr>
<tr>
<td>IF (Input port)</td>
<td>5-32</td>
</tr>
<tr>
<td>Keys</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>Keywords</td>
<td>2-3</td>
</tr>
<tr>
<td>LEN</td>
<td>5-14</td>
</tr>
<tr>
<td>LET</td>
<td>4-2</td>
</tr>
<tr>
<td>LF Line feed key</td>
<td>1-3, 3-2, 3-6</td>
</tr>
<tr>
<td>Line numbers</td>
<td>2-4</td>
</tr>
<tr>
<td>LIST</td>
<td>3-15</td>
</tr>
<tr>
<td>LL - Line length</td>
<td>3-1, 5-32</td>
</tr>
<tr>
<td>LOG</td>
<td>5-5</td>
</tr>
<tr>
<td>LOG10</td>
<td>5-5</td>
</tr>
<tr>
<td>Logical operators</td>
<td>4-17</td>
</tr>
<tr>
<td>MAT</td>
<td>7-2</td>
</tr>
<tr>
<td>Mathematical functions</td>
<td>5-5, A2-1</td>
</tr>
<tr>
<td>Matrix operations</td>
<td>7-1, 7-4</td>
</tr>
<tr>
<td>Merge (APPEND)</td>
<td>3-16</td>
</tr>
<tr>
<td>ML (Memory limit)</td>
<td>5-18, 5-2</td>
</tr>
<tr>
<td>MODE SELECT key</td>
<td>3-2, 3-3, 3-6, 3-1</td>
</tr>
<tr>
<td>Monetary format</td>
<td>5-27</td>
</tr>
<tr>
<td>NEXT</td>
<td>4-12</td>
</tr>
<tr>
<td>NOT</td>
<td>4-17</td>
</tr>
<tr>
<td>Number keys</td>
<td>3-2, 4-5</td>
</tr>
<tr>
<td>ON...ERRETE</td>
<td>5-30</td>
</tr>
<tr>
<td>ON...EXIT</td>
<td>4-14</td>
</tr>
<tr>
<td>ON...GOSUB</td>
<td>5-4</td>
</tr>
<tr>
<td>ON...IF</td>
<td>4-11</td>
</tr>
<tr>
<td>ON...RESTART</td>
<td>4-8</td>
</tr>
<tr>
<td>OP (output port)</td>
<td>5-32</td>
</tr>
<tr>
<td>OR</td>
<td>4-47</td>
</tr>
</tbody>
</table>

OUT

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAUSE</td>
<td>4-10</td>
</tr>
<tr>
<td>PEK</td>
<td>4-2</td>
</tr>
<tr>
<td>POKE</td>
<td>6-2</td>
</tr>
<tr>
<td>POS (0) - Position</td>
<td>4-33</td>
</tr>
<tr>
<td>PRINT</td>
<td>4-3, 5-20</td>
</tr>
<tr>
<td>Program</td>
<td>2-6, 3-1</td>
</tr>
<tr>
<td>Program storage</td>
<td>3-11, 3-13</td>
</tr>
<tr>
<td>Punctuation</td>
<td>1-3</td>
</tr>
<tr>
<td>READ</td>
<td>4-6, 5-21</td>
</tr>
<tr>
<td>Recorder</td>
<td>3-9</td>
</tr>
<tr>
<td>Relational operators</td>
<td>4-16</td>
</tr>
<tr>
<td>REM</td>
<td>4-1</td>
</tr>
<tr>
<td>REN</td>
<td>3-5</td>
</tr>
<tr>
<td>REPEAT Key</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>RESTORE</td>
<td>4-8</td>
</tr>
<tr>
<td>RETURN (Statement)</td>
<td>5-5, 5-7</td>
</tr>
<tr>
<td>RETURN key</td>
<td>3-2, 3-6</td>
</tr>
<tr>
<td>REMIND</td>
<td>5-2</td>
</tr>
<tr>
<td>RMD</td>
<td>5-5</td>
</tr>
<tr>
<td>RUN</td>
<td>3-7, 4-9</td>
</tr>
<tr>
<td>SAVE</td>
<td>3-12</td>
</tr>
<tr>
<td>Scalar operations</td>
<td>7-3</td>
</tr>
<tr>
<td>Scientific format</td>
<td>5-12</td>
</tr>
<tr>
<td>SCRATCH</td>
<td>3-4, 5-32</td>
</tr>
<tr>
<td>BRANCH</td>
<td>5-12</td>
</tr>
<tr>
<td>Semi-compiled</td>
<td>4-4, 5-27</td>
</tr>
<tr>
<td>SET</td>
<td>3-11, 3-13</td>
</tr>
<tr>
<td>SETC</td>
<td>5-31, 5-31</td>
</tr>
<tr>
<td>SIN</td>
<td>4-12, 4-12</td>
</tr>
<tr>
<td>SIN</td>
<td>5-5</td>
</tr>
<tr>
<td>SINH</td>
<td>5-6</td>
</tr>
<tr>
<td>Space bar</td>
<td>3-3-4-5</td>
</tr>
<tr>
<td>SQR</td>
<td>5-5</td>
</tr>
<tr>
<td>Statements</td>
<td>2-3, 5-31, A1-3</td>
</tr>
<tr>
<td>STOP</td>
<td>4-9</td>
</tr>
<tr>
<td>STR</td>
<td>5-15</td>
</tr>
<tr>
<td>Strings</td>
<td>5-5, 5-9</td>
</tr>
<tr>
<td>Substrings</td>
<td>5-13</td>
</tr>
<tr>
<td>T (text)</td>
<td>3-11</td>
</tr>
<tr>
<td>TAB</td>
<td>4-5A</td>
</tr>
<tr>
<td>Text storage</td>
<td>3-11, 3-13</td>
</tr>
<tr>
<td>TAR</td>
<td>4-6</td>
</tr>
<tr>
<td>THEN</td>
<td>4-18</td>
</tr>
<tr>
<td>THOOP</td>
<td>5-19</td>
</tr>
<tr>
<td>THOS</td>
<td>5-19</td>
</tr>
<tr>
<td>TIPO(0)</td>
<td>4-7</td>
</tr>
<tr>
<td>VAL</td>
<td>5-15</td>
</tr>
<tr>
<td>Variables</td>
<td>2-5</td>
</tr>
<tr>
<td>Variables, string</td>
<td>5-10</td>
</tr>
<tr>
<td>WAIT</td>
<td>6-3</td>
</tr>
<tr>
<td>XEQ</td>
<td>2-1, 3-15</td>
</tr>
<tr>
<td>Z</td>
<td>5-25</td>
</tr>
<tr>
<td>ZER (Zero matrix)</td>
<td>7-2</td>
</tr>
</tbody>
</table>

Continued on page 11.
Dear Stan:

NORTHSTAR ERROR FIX

DOES I goofed. In my program for loading North Star files (PROTEUS v. 2, no. 8, July-August, 1979), line 77 now reads:

LI X 5200H

It should read:

LI X 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 inadequacies, 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 bugs. Better works 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,

[Signature]

THINKER TOY 20

Dear Stan:

I noticed in your issue (May-June) that you were planning to buy a "Thinker Toy 20" system.

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

If you reset the Sol with CPM in RAM, you are not able to jump back into CCP because of a hardware error in Thinker Toy...unless you first change location zero 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 500K bytes when used with CPM which I also purchased. You get "7" tracks of 20 sectors, each with 256 bytes, or 512,576 bytes.

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

A good point for "Thinker Toy" is that the 801 tool kit is at 0000 to 07FF to start things going, which permits anyone with a Northstar disk system to keep it up, and use either.

Sincerely,

Bernard Pickton, Jr.
Renfro Hotel
3120 Collins Ave.
Miami Beach, Florida 33140

ROY HEYBROCK
CERTIFIED FINANCIAL PLANNER
212 S. Person Street
Greenfield, North Carolina 27403

Dear Stan:

First of all I want to thank you for your efforts in putting together 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 PCF 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 magazines about it. I bought Asborne & Associates "880A/885 Assembly Language Programming" but couldn't find any reason to use 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 in ASB/R doesn't seem 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 accessing that data with each sub program. To do this you can have options for the operator and the program will then "XEO XXX". The XXX is the name of your branch to program. You can then read back the necessary data, make changes or whatever and then "XEO YYYY", where YYYY is the name of your original or main program. I find this very useful 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 program to SET UP = PRINT instead of 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 seem to have to enter the same information over and over.

Sincerely,

ROY "PLANNING FOR FINANCIAL INDEPENDENCE"
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 some machine combinations represented. We have Sol/Novell, Sol/NorthStar and of course, Sol/cassette.

We have had some of the same problems in dealing with Proteus as other users have had and appreciate their solution. Finally we got the rest of them. In conversation with the people at ForTek, I found that PTGO is bankrupt. I trust this is the truth since there result in a “line disconnected” response.

Most of us feel it really won’t present a problem since we have not had problems we can’t fix in over two years. All of our problems in the Sol are for instance have either been in the power supply or in the failure of the head positioning. In the Sol, most problems have been in the memory boards. We have had PROTEUS as well as PTGO software, and some more utilities from another company near here.

One of our local users has written a test program for use with files provided by PICT or EDIT S.0. We have both cassette and disk versions and we have special drivers for the NEC S98N and similar systems.

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

If any of the Proteus users care to have one of our information, I hope they will write me at the address below. Keep on the good work, I am looking forward to the next issue.

Sincerely,

Bill Anderson
415 Forest Ave. North
Fargo, ND 58102

QUESTIONNAIRE LACKING

Dear Proteus,

I believe your questionnaire 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 combinatorial tape and disc system is the way to go.

B.W. Littlejohn
Box 58, 861
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 luck 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.

EDITOR: Tony package used to sell for $25 and $275 respectively. Will anyone who has tried it read it?
Enclosed is my PROTEUS membership application for 1980 and a check for $63: $58 for dues and $6 for a QST membership. Also enclosed is source code for a set of changes I have made to Extended BASIC, and an example of the changes I have made.

I know that you are capable of compiling the code with PROTEUS, but I am not sure that you can compile the source code. I have not made any assumptions about the source code, and I have not included any comments or annotations. The source code is self-contained and includes all necessary documentation.

I have just completed a disk operating system which may be of use to some members. Your first reaction may well be, "Why another one?" I wrote it because I wanted to run existing software as little change as possible, and with all of the functions of the SOLARIS operating system. Therefore, I simply extended SOLARIS and added two new unit numbers, two new commands, and one new SET command in order to add two simple density, soft sectored, mini-flourescent disc drives. With some pride, I call my extension SOLARIS Bolton's Disc Operating System, or DOS. 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 typing alone. Back to DOS. BOOS has some hooks into SOLARIS, so the SOLARIS EKROM must be rewritten to use BOOS. At the other end, BOOS talks to the disc drives through calls to $10 for GPM. BOOS is stuffed into the system KAM at G000. However, there is still room for a good sized stack there. Since the listing alone for BOOS in 30 pages, I am not going to send it in unless someone is interested in using it.

My drives are Shourt model 5A400. They are 26 drives and power supply are in a custom cabinet 17" wide, 9" deep and 6" 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 drive controller is the Versaflow by S.O. Systems. On that board, I found the Wait State Generator would not work with the SOL. By cutting the connection to CI26 at 2300 on the back of the board and connecting a 3dB output (pin 38 of PD11718) to that pin, I provided the same function in software by inverting the state of bit 6 on input port X3 with no loss in speed. My firmware is in 216s on the Cramen 32K Byesaver board. I have had no problems at all with it. Programming the 2716s (815/EL) is very easy. Furthermore, the 2708 Ferocessory Module is very easily changed to accept a 2716s where the 815s busses 85/5 cut traces between feedthroughs 21 and CO and 19 and 512; but jumper between feed throughs 51 and 19. The take place in socket U1. The changes provide 5v to pin 21 (9pp) and connects the 700 series of input to pin 19 (1A0).
12E6 0720 * files may be opened. However, a variable indicating
12E6 0730 * the address of the current PCB at 1600-1601 must be
12E6 0740 * saved. The next few orders do that.
12E6 0750 *
12E6 0760 * Initialize 2 new PCBs to closed.
12E6 0770 ORG 156FH
12E6 0780 DS 90
12E6 0790 ORG 1613H
12E6 07A0 DS 0
12E6 07B0 ORG 1627H
12E6 07C0 FF 0830 DS 3FH
12E6 07D0 * Relocate the addr of current PCB to 162A-1629
12E6 07E0 * (future patch area remains at 162A-1840)
12E6 07F0 162C 0080 ORG 970H
12E6 0800 162D 0187 Dw 1628H:1m SAVE routine
12E6 0810 162E 0080 ORG 020DH
12E6 0820 162F 0188 Dw 1628H:1m SEQ,APPEND,GET routine
12E6 0830 1630 020D ORG 1358H
12E6 0840 1631 0010 Dw 1628H:1m locate PCB routine
12E6 0850 1632 0720 ORG 1330H
12E6 0860 1633 0030 Dw 1628H:1m move PCB to header routine
12E6 0870 1634 0040 ORG 133EH
12E6 0880 1635 0050 Dw 1628H:1m move PCB to header routine
12E6 0890 1636 0000 ORG 144DH
12E6 08A0 1637 0020 BB 4 ;Allow unit numbers up to 4
12E6 08B0 1638 0030 *
12E6 08C0 1639 0040 * The following modes the REMIND routine to delete the messages
12E6 08D0 163A 0050 * and to set rex. E-access requested when calling FOPEN.
12E6 08E0 163B 0060 * Up to 1493, the file has been closed (to SOLOS).
12E6 08F0 163C 0070 ORG 1693H
12E7 0493 21 C5 15080 LDX H.15C4H
12E7 0494 21 9E 1090 MOV A,M :=unit 9
12E7 0495 21 C3 1100 INX H
12E7 0496 21 5E 1110 MOV E,M :=E-access requested
12E7 0497 21 2E 1120 INX H
12E7 0498 21 2F 1130 INX H :HL points to header
12E7 0499 21 C3 1140 JMP 148CH ;There we call FOPEN
12E7 049A 21 1E 1150 * (future patch area available from here through 148B)
12E7 049B 21 80 1160 *
12E7 049C 21 70 1170 * The following eliminates messages & E-access requested
12E7 049D 21 80 1180 * for BASIC's main byte file operating routine.
12E7 049E 21 90 1190 ORG 1568H
12E7 049F 21 FE 0120 CPI 3
12E7 04A0 21 DD 0110 JNC 29FCFh :JAC error if access requested >2
12E7 04A1 21 D7 1220 ORA A
12E7 04A2 21 CA 0290 JS 29FCFh ; or =0
12E7 04A3 21 82 1240 STA 15CH
12E7 04A4 21 7F 1250 MOV W,A :=E-access requested
12E7 04A5 21 3F 1260 JMP 15AH :Finish opening the file there
12E7 04A6 21 3F 1270 * (future patch area from 1581 through 15A4)
12E7 04A7 21 7F 1280 *
12E7 04A8 21 7F 1290 * Make HL point to name in PCB & unit no.
12E7 04A9 21 7F 12A0 * when closing files of a crashed program.
12E7 04AA 21 7F 1300 ORG 37H
0032 2D 78 211353 CALL CRSFY
0035 1330 *
0036 1340 ORG 1578H
1578 23 1350 ORSFY INX H
157C 7E 1360 MOV A,M :=unit no.
157D 11 00 1370 LXI 0,3 :Set up to point HL at name in PCB
1580 09 1380 KEY
1581 1390 *
PROGRESS REPORT FROM YOUR PUBLISHER

Source code purchase: No news to report. It's still 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, ThinksToys. Micromation is all talk and no action as far as I am concerned; Delta Products has incompatibilities with Sol; a tech from DateSpeed is coming 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 OPER

After talking to many of the PROFEUS 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 to do if your system goes down. 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 gas noisy boards. (Data devices expected in 1-2 months)

My position would be one of the representative of the owner of the equipment to be repaired. I would need your equipment to someone who I felt was qualified as a repair person and had reasonable rates. Personally, I would be happy to help all the other systems and have had to seek repair from these people myself. I have found places that repair to the PERSCl disks but not to the tolerances I would need desperately. Helios, (did you know that most PERSCl repair people adjust the PERSCl to operate in a "soft sectored" configuration. The Helios uses 1/2 hard sectored disks and the tolerances are much smaller, so small, in fact that the PERSCl can be operating perfectly in another system but will fail when connected to a Helios and most PERSCl repair people do not have a test disk 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.C.'s, in lieu 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.

If something like your Helios goes down, call me. I will try and help you at the cost of my equipment (what is in my stock). I will also give you an estimated price as to the cost to repair your item. You need to me (after packaging the equipment nicely) and pay me the amount of the repair up front. The $50 will be my fee. I will then find the repair person who can get 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 will check it up when finished. I will then "sell" the piece of equipment on my system to make sure it works. That way any problem can be worked out without you having to wait and pay shipping costs unnecessarily.

All the repair people I will deal with will have Sol equipment with PerScI drives. If there are no Sol costs, I will still help you at the cost of my equipment. If there are any Sol costs, I will still help you at the cost of my equipment. I will also help you at the cost of my equipment.

"This way you have someone who is in 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 the equipment costs can be charged to me instead of you. Please feel free to call me if you have any questions at 98-4445 or drop me a line at 137 Highlane ave, Vacaville, Ca. 95688

Note: This is a function being provided by "Cory" Data Service and not an official PROFEUS project."
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 CONST, CONIN, and CONOUT routines are included in SOL. SOLO splits the screen into two sections. The upper section is fourteen lines in length, and the lower section is sixteen lines in length, and it also supports a RUN IN模式 in which many command lines may be written as a single COMMAND LINE.

When a carriage return is input, the COMMAND LINE is made available in BASIC after the calling routine. Until this terminating carriage return is hit, the entire 128 character COMMAND LINE can be edited using the facilities built into SOL. 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 spaces (tabs code spaced 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 retrying 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 so that all 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" DMM single density disk, soft sector; 8" double density, soft sector; 5 1/4" single density disk, soft sector. Microplex 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 density 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 am 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 McGhee S.D.B.
10 PRINT CHR$(11)
20 REM 'CHGBAG'
30 REM PROGRAM WILL CHANGE THE BASE OF ANY NUMBER INPUT
40 REM AS DESCRIBED IN THE PRINTED INSTRUCTIONS FOLLOWING
50 REM
60 REM COPYRIGHT BY JAC W. RANDOLPH
70 REM FEBRUARY 1979
80 REM RANDOLPH & ASSOCIATES, INC.
90 REM 586 SHADES CREST ROAD
100 REM BIRMINGHAM, AL 35226 PH 205-822-2339
110 REM
120 "THE FOLLOWING PROGRAM WILL CONVERG ANY NUMBER INPUT
130 "WHETHER BASE 8,10 OR 16 (OCTAL, DECIMAL OR HEX)."
140 "IF DECIMAL NUMBERS ARE INPUT, THE HEX AND OCTAL"
150 "EQUIVALENT WILL BE GIVEN. IF HEX IS INPUT, THE DECIMAL"
160 "AND OCTAL EQUIVALENT WILL BE GIVEN; ETC."
170 "AN 'H' FOR 'HEX' AND AN 'O' FOR 'OCTAL' MUST FOLLOW"
180 "HEX AND OCTAL VALUES.... THIS IS THE ONLY REQUIREMENT"
190 "!
200 !
210 INPUT "INPUT NUMBER YOU WISH TO CONVERT: ",X
220 A$="ABCDEF"
230 M$=X$(LEN(X$))
240 REM.. LAST CHARACTER
250 REM ASSUME DECIMAL IF NOT 'D' OR 'H'
260 IF M$="D" THEN 320
270 IF M$="H" THEN 690
280 IF M$="O" THEN 870
290 IF '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$)
340 IF X$(I)<="9" THEN 360
350 GOTO 370
360 IF X$(I)="9" THEN 400
370 IF X$(I)="A" THEN 400
380 "NO DECIMAL NUMBERS PERMITTED LESS THAN ZERO"
390 "OR GREATER THAN SEVEN. IF OCTAL NUMBER"
400 GOTO 210
410 NEXT
420 J=8
430 GOSUB 440 C=I
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)="9" THEN 580
490 M$=0
500 FOR K=1 TO 7
510 IF X$(I)="A" THEN 530
520 X$=X$(I)
goto 590
530 M$=M$+1
540 NEXT K
550 IF M$=6 THEN 590
560 "INVALID HEX CHARACTER ",X$(I),"
570 GOTO 210
580 X$=VAL(X$(I))
590 IF S>1 THEN 610
600 S=S+X$(I)
goto 640
610 X$=I
620 FOR M=1 TO 5
630 X$=X$+I
640 NEXT M
650 S1=S1+X$X$1
660 GOTO 450
670 PRINT "DECIMAL VALUE IS ",S1
680 RETURN
690 FOR I=1 TO LEN(X$)-1
An Instruction Decoder for the ALS8 Simulator
by Joe Maguire
P.O. Box 3742 OS
Anchorage, AK 99518

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
read the instruction mnemonics corresponding to the machine
code being displayed. A session with the Simulator required
constant referral to the 8080 instruction set.

Realising 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 mnemonics 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 RST instruction at
address F735H from the stack)

A sample display is given below:

- C04 00000 0 0 0 0 0 0 0 0 21 0000
  C3 0 0 0 0
  C9 0 0 0 0
  MP 0
- C1C 00000 0 0 0 0 0 0 0 0 0 0 0 0 21 21
  CB 0 0 0 0
  BF 0 0 0 0
  BF 0 0 0 0
  0 0 0 0
  BF 0 0 0 0
- C1D 00000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 21 21
  CB 0 0 0 0
  BD 0 0 0 0
  BD 0 0 0 0
  0 0 0 0
  BD 0 0 0 0
- C2F 00000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 21 21
  BF 0 0 0 0
  C 0 0 0 0
  0 0 0 0
  C 0 0 0 0

Gordon Wilson
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Address</th>
<th>Flags</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1660 TQ3 MOV A, C</td>
<td>1670 ADD A</td>
<td>A,C</td>
<td></td>
</tr>
<tr>
<td>1680 MOV A, D</td>
<td>1710 DAD B</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>1730 CALL PRINT</td>
<td>1740 POP PSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800 D7 CALL RDBYT D7</td>
<td>1810 MOV A, D</td>
<td>D,A</td>
<td></td>
</tr>
<tr>
<td>1820 CALL RDBYT</td>
<td>1850 CALL ASCII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1850 CALL JMP ASCII</td>
<td>1910 LXI H,TAB-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1910 MOV A, C</td>
<td>1920 DAD B</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>1950 D8 CALL RDBYT</td>
<td>1960 JMP ASCII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1970 MOV A, C</td>
<td>1990 ADD A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>1990 MOV A, D</td>
<td>2000 ADD A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010 MOV A, C</td>
<td>2020 MOV A, D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2030 ADD B</td>
<td>2040 JMP PRINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2050 MOV A, C</td>
<td>2060 DB 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2070 MOV A, D</td>
<td>2080 TABLE DB 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2090 MOV A, C</td>
<td>2100 DB 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2110 MOV A, D</td>
<td>2120 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2130 MOV A, C</td>
<td>2140 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2150 MOV A, D</td>
<td>2160 DB 7FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2170 MOV A, C</td>
<td>2180 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2190 MOV A, D</td>
<td>2200 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2210 MOV A, C</td>
<td>2220 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2230 MOV A, D</td>
<td>2240 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2250 MOV A, C</td>
<td>2260 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2270 MOV A, D</td>
<td>2280 DB 1FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2290 MOV A, C</td>
<td>2300 MOV A, D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2310 MOV A, C</td>
<td>2320 MOV A, D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2330 MOV A, C</td>
<td>2340 MOV A, D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2350 MOV A, C</td>
<td>2360 MOV A, D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2370 MOV A, C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
North Star Library Forming

We have one person at present working on some of the software we have for the North Star/Sol 20 pair. The work consists of sorting through "0" 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 similar 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 Severs at 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 can't 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 lured me to the task, but the I'm looking for. I didn't receive it yet, but it'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 you have 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 disk and upgrade from 300 baud to 1200 baud. If you want, I'll evaluate it and send you a camera ready report on it and easy (difficult) it was to connect.

Thanks

Ron Genova
501 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 cassette Sol software to memory 2000. (North Star D.O.S. resides from 2000 to 255F.) 4. Type EXEC 2020 (This will put you into N.S. D.O.S.) 5. Create your file name and length of file type isn't. Starting address needed for the Sol 5. WD 2000 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.
from the author or copyright holder giving the library permission to distribute 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 assigned 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 the demand for cassettes than the cassette itself 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 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 BASICS internal compiled format, i.e. SAVE name
2. PTCO ECRBASICS internal compiled format, i.e. SAVE name
3. PTCO ECRBASICS text format, i.e. SAVE name
4. Assembler source files as memory images of ALS-8 or SOTA 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 byte length and the trailing CRs.
   B. The two-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, 16H.

   This file format is written by the SAVE commands of the newer versions of SOTA, and by using SUGS/CIVER to save a memory image of an ALS-8 file or an older SOTA file.

5. Assembler source files and BASIC programs as ASCII text in the PROFEUS STANDARD BYTE ACCESS tape format, described below. The PROFEUS STANDARD BYTE ACCESS format is intended to be the universal bridge between all of the various disk systems. Pico 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.

6. Executable object code, designed to work under SOLUS/CIVER. Where possible, SOLUS/CIVER should be accessed only through the entry using tag table, if necessary to access byte access tape, or 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 DISK/APC program written by Richard Greenlaw, and other disk systems of cassette Cs. So many of our readers 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.
B. 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 SULS or our compatible computers; we don't want to go into competition with the CP/M user group.

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 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 well 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
276 Kieneville Ct., NC
Conyers, GA 30013

If you wish a personal reply, please enclose a self-addressed stamped envelope (SASE). A catalog showing contents of all of the library cassettes should be available in January, 1989. 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 transferred to Tony Severa. Disk H-4 can be ordered by contacting Tony at 111 Hickory Hill Ave., Valdosta, Ga. 31601. Disk H-1 thru H-3 can still be ordered by writing to Stan Sokolow at Proteus until January 1988. 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 $5 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 program sent 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, PTDDO, 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 rep's 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' 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 PTDDO file type will indicate the language.)
   :O Object code, such as compiled form of BASIC or PASCAL. (The actual object language is indicated by 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 similar to the :D but very brief. This file will be published in the catalog of library programs.
   :N 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 file names: PROG:O:O:O:O, 10:O:O:O, 10:O:O:O, etc. A device driver ("DEVICE") submitted as both source and assembled files of type "D" will need files: DEVICE:O:O:O:O, DEVICE: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 program. Whenever a PFC file type convention is relevant, it should be used. (See section of the PTDDO 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 ;0 System files (reserve for PTDDO)
81 @1 Numerical data in binary form.
82 @2 Numerical data in BCD form.
83 @3 Stored FOCAI program.
84 @4 Semi-compiled BASIC/5 program.
85 @5 Semi-compiled EDBASIC program.
86 @6 Source (text) EDBASIC program.
87 @7 Serial Access File.
88 @8 Random Access Data file.
C1 C Archive (SAVE) file.
AE @ Default.
A4 S DC file with command lines.
D4 T Text file (also BASIC/5 text form)
C8 P PASCAL source code (text form)
P8 P PASCAL p-code form (semi-compiled)
00 @ Image files associated with system
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 8080 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>EXECBASIK.LNK

The program would initially do the following five things:
1. Load and execute at the CP/M transient address of IVEH.
2. Relocate itself out of the way, perhaps in the GWIB range.
3. Find and read in at address W the specified disk file.
4. Load registers HL with the address of a surrogate jump table which would replace the usual SLLCB/CLIB jump table, and
5. Begin the program by a jump to address W.

The surrogate jump table should interface all, or at least some, of the SLLCB/CLIB functions to the CP/M system. The input routines (SINF and AINF) could simply jump to the appropriate points in the SLLCB/CLIB jump table. The output routines could send Pseudos 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 .BIA be used for this purpose. A jump to SLLCB should re-boot CP/M. Eventually, sequential disk I/O routines could be written to simulate the SLLCB/CLIB byte access tape access routines.

If you have implemented all or part of a SLLCB/CP/M interface, please write to:
Proteus Cassette Software Library
C/O Lewis Roseley, Jr., Librarian
2576 Glenola Ct. N
Conyers, Ga. 30013

THANKS.

PROPOSED STANDARD FOR BYTE ACCESS TAPE FILE EXCHANGE

When our user's society (Proteus, formerly known as SLLCB) first began, most of us were using the SLLCB/CLIB byte access tape format for most of our programs and data. Because of this, a common link between all SLLCB owners (and compatible CP/M systems operating under SLLCB), software transfer was simple and easy. However, at this time (late 1973) most of us are using some kind of disk system, and because of this, block access tape is not a universal exchange medium. Unfortunately, it is impossible to completely use a compatible internal block tape format. To help overcome this obstacle, the library is proposing the PROPOSED STANDARD FOR FILEMAKERS. 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 typewriter".

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 SLLCB/CLIB, and would intercept a listing of the program, and send it to the tape instead of the screen. The WRITE driver should filter out all 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 input in 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 of the system should provide them. This would usually be done by the system automatically if the READ driver properly simulates the "fast typewriter". The READ driver should handle end-of-line and tape-read-error conditions, typically by giving a message and switching back to the keyboard for further input.

This format is compatible with the byte access format used by ECHIAC, PICLIC, and the PICL PACK and UNPACK utilities. Currently, compatible drivers have been written for Micropolis (by Kelvin Walton) and ABU-8 (by Lewis Roseley, Jr.) drivers for CP/M which 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 compatible drivers and software. Address all correspondence to: Proteus Cassette Software Library
C/O Lewis Roseley, Jr., Librarian
2576 Glenola Ct. N
Conyers, Ga. 30015

If you would like a personal reply, please enclose a BASE.

BEKES READING AREA COMPUTER ENTHUSIASTS MEET

This group has been meeting for quite some time and welcomes anyone living in the area to contact them for information on upcoming meetings:
Paul Adams, Coordinator
Box 210 A Rd. #4
Reading PA 19405
215-929-2637

Ron Groves, Secretary
601 Friedensburg Rd.
Reading, PA 19406
215-775-5018

Bob Dietzel, Treasurer
4301 Club Drive
Reading, PA 19406
215-775-7240

/
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 1739, Irvine, CA 92713 has purchased the entire Processor Tech inventory 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 $125.50, those beautiful SOL keyboards for $139.95, and 8K RAM assemblers for $129.95. Not only this, but Mr. Cassity said that they were interested in continuing support on some of the SOL products if you specifically ask for it, they will send you a copy of their complete inventory list of Processor Tech parts. I looked over the list and have 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 KIGS 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 driven density controller at present, plus a Micromation double drive which I use in both single and double density on S° 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 nine to add lots of extra features like the ability to change I/O drivers at any time using a control word. The original BIOS was immediately usable 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, COMIN, and CONOUT routines for use with the ELECTRIC PENCIL or 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. So 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 McGhee
Don Bosco Tech
202 Union Ave.
Pateros, N.J. 07502

---

FOR SALE:

The Maria Computer Center is selling six (6) of its SOL20 Terminal Computers with 24K and Hitachi monitors for $1600 each or $7500 for all. They are in excellent condition - we need computers with color graphics. Call David at (415) 472-8550 or (415) 892-8846 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 PTOS DISKETTE TO PTOS 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 1.7 MOD 2...........................................................$10

PROPRIETARY 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 BGC. SEE DESCRIPTION IN PROTEUS NEWS VOL2, #6..........

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

P3.. MAILSORT SUPPLEMENT TO MAILMASTER, BY BGC. 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 BGC. SEE VOL 2, #6...$500 / 35 MANUAL

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

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

P7.. ACCPAC PROGRAMMER'S PACKAGE FOR CREATING CUSTOM ACCPAC
PROGRAMS BY BGC. 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............................$5995 / 200 MANUAL

CASSETTE LIBRARY CASSETTES

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

ORDER FROM "LEWIS MUSELL, JR. 2576 GLENDALE COURT NE, CONVYER, GA 30096"

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.. 6060 Debugger Users Manual No more.
D3.. Extended Disk Basic Users Manual Loan only
D4.. SDM-1 Display Module Users Manual No more.
D5.. Cassette Focal Users Manual No more.
D6.. Advanced 6800 Assembler Users Manual Loan only
D7.. Basic/5 Users Manual Loan only
D8.. SOL Terminal Computer Users Manual No more.
D9.. Software #1 Manual No more.
D10.. ASCII Basic Software #2 Source Listing No more.
D11.. SOLOS Cuts Users Manual Loan only
D12.. Cuts Computer Users Manual Loan only
D13.. 6PN and CPN Sol Assy and Test Instr. No more.
D14.. Subsystem #9 Users Manual No more.
D15.. Cassette Pilot Update 731060 $5.55
D16.. Cassette Pilot Update 731069 $5.55
D17.. Software #1 Update 731107 $4.50
D18.. Extended Disk Basic Update 731062 $4.70
D19.. SDM-1 Update 731063 $4.70
D20.. Extended Cassette Basic Update 731064 $1.50
D21.. EXT. Disk Basic Update 731065 $4.55
D22.. SOLOS Chess Update 731043 $2.03
D23.. PTOS Update 731072 $1.50
D24.. PTOS and Wordwizard Update 731074 $1.58
D25.. Wordwizard Update 731075 $1.58
D26.. 32KRA Users Manual Loan only
D27.. 32KRA Users Manual $5.55
D28.. 32KRM-1 Users Manual Loan only
D29.. ACCESS, MARCH 1977.............................................$1
D30.. THE SMALL COMPUTER CATALOG........................................$1
D31.. SOL Manual Addendum 1 & 3...........................................$1
D32.. SOL Manual Addendum 1 & 3 Loan only
D44.2ENO "CHANGE #" $1.50
D45.3LO MANUAL CHANGE #9 $1
D46.3LO MANUAL CHANGE #10 $1.50
D47.3LO MANUAL CHANGE #11 $1.50
D48.3LO MANUAL CHANGE #13 $1.50
D49.3LO MANUAL CHANGE #14 $1.50
D50.3LO MANUAL CHANGE #16 $1.50
D51.3LO UPDATE "31049" $1.50
D52.2ASSY PHC CHANGE #850 $1.50
D53.4KKK ASSY AND TEST INTR. $1
D54. ASSY CHANGE #6-2 REV C $1.55
D55.5LO HYKFILE INTERFACE UPDATE "31075" $1.55
D56.6KRA INSTALLATION GUIDE $1.50
D57.6KKKA E2KKK UPDATE "31066" $1.50
D58.7KKRA USERS MANUAL $1
D59.7KKRA & 3KKRA UPDATE "31024" $1.50
D60.7KKRA DIAGNOSTIC TEST $1
D61.7KKRA 1CHGE #2 $1.50
D62.7KKRA & 3KKRA UPDATE "31041" $1.50
D63.7KKRA & 3KKRA UPDATE "31042" $1.50
D64.HELIO UPDATE "3106" $1.50
D65.7DDOS UPDATE "31073" $1.50
D66. PARA SOL DEBUGGERS USERS MANUAL $1
D67. HELIOS UPDATE "11040" $1
D68. HELIOS UPDATE "51011" $1
D69. FILD RETROFIT NOTICE 4/1/79 HELIOS $1
D70. FILD RETROFIT NOTICE 4/19/79 HELIOS $1
D71. PERSONAL PCB ASSY LOGIC SCHEMATIC $1
D72. HELIOS II DISK SYSTEM MANUAL $2.50
D73. HELIOS II MANUAL CHANGE #3 $2.50
D74. HELIOS II MANUAL CHANGE #4 $2.50
D75. REV. A OLD LOOSE LEAF HELIOS MANUAL $1.75
D76. WORDWIZARD UPDATE #3 $2.00
D77. S138 NEWSLETTER #11 $2.00
D78. CHAPTER VIII FOR SOL SYSTEM MANUAL $2.00
D79. DIALO SERIES 2300 MAINT. MANUAL $2.00
D80. DIALO SYSTEM MAINT. AND PRICING MANUAL $2

PLEASE ADD $1.00 PER COMPLETE ORDER TO COVER HANDLING AND POSTAGE. MAKE CHECKS PAYABLE TO "TANY SEVERA" AND MAIL TO 115 HIGHLAND AVE, VACAVILLE CA 95687 (ON LOANS CALL 707-424-6447 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 Vol 2 #4 for more details on these items)

WORLD GENERATOR by Tony Severa

120 REM PROGRAM TO MAKE RANDOM WORLDS.
121 REM PROGRAM IS MEANT FOR THOSE OF YOU WHO ARE INTO D & D OR
122 REM FANTASY ROLE PLAYING GAMES. THIS PROGRAM, WHEN RUN WILL DESIGN
123 REM FANTASY WORLDS FOR GAMES LIKE TRAVELER, GAMMAMWORLD OR D&D.
124 ENJOY
200 REM
300 REM MAP LOCAL SUBSECTOR
400 REM MAP SPACE PLANET
500 REM MAP SPACE LAKES: JUMP ROUTES
600 REM MAP PORT TYPES
700 LET $S$M$H$E$(O)*3*1
TABLE OF CONTENTS

Editorial.................................................. 1
Cassette News........................................... 2
EPROM Services Being Provided....................... 3
Repair Services Available............................. 3
Floppy Disk Note........................................ 3
NAAK ROM Information................................ 3
Writing END Files on Cassette........................ 4
Index to ext.Cassette Basics......................... 4
North Star Error Fix.................................... 4
Thinker Toy Users Opinion............................ 5
Novice Corner........................................... 6
PROTEUS Needed Even More............................ 6
Response to Questionnaire............................. 6
Search for Color Graphics Board for Soli........... 6
COLOR Prices Slashed................................. 6
Modifications to PCPC EBCASIC......................... 6
Service Center Opened................................ 7
SOL for CP/M Users................................... 7
Base Change Program................................. 8
Z80 or WAMCO 8K "Trouble Shooters"................. 8
Instruction Decoder for ALGOL Simulator............ 9
North Star Library Taking Shape..................... 9
North Star Help Needed............................... 10
Cassette Library Guidelines.......................... 10
Helios Library Changes Hands......................... 12
Helios Library Guidelines............................ 12
Interfacing CP/M with PDOS............................ 12
Proposed Standard for Byte Access Tape Exchange... 12
More Info on Advanced Computer Products........... 14
Driver for 1610/1620 Printers......................... 15
PROTEUS Price List................................. 16
D & D World Generator Program....................... 16
Lunar Lander Modified for Real Time................ 17
Classifieds............................................. 17

Continued from Page 1.

High School their status as to what type of micro they have and if they have a Sol, turn them on to us and/or turn us on to them. We'll need them information on what we are doing and maybe they can share none of their student written software with the rest of PROTEUS members.

When I picked up all the literature desired 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 more little article can help someone else. Please read in any ideas, articles, tutorials, programs and reactions to us for inclusion in the newsletter. Don't worry, if you don't want it attached to it, we'll be glad to omit it. Main thing is that you are sharing your knowledge with someone else.

Notify me of local repair stations for P.T. equipment and try to include price lists. If you want to sell or buy equipment please notify us for inclusion in your classifieds column. It's a free service to all.

I see a need for more articles oriented towards the beginner or novice especially around the use of PDOS and its many capabilities. Here are some 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 has been changes to PDOS 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, Fossil and 807 with simple examples of how they work. I would also like to see more in the way of examples of how to use noroisribbon. Please use a new ribbon. Machine-readable articles should be compatible with Solos, Cuters, PDOS 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
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 #0 in memory location A890 (the location is A881 for PTDOS 1.4; from now on PTDOS 1.5 will be assumed) before using REAPR. This can be accomplished by exiting to SOLOS and using the ENTER command, 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., 9:10 POKE 43152,0: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 “ZAPS”. I’m lazy, so I would commence step 1 by typing “EDIT ZAPS: <cr>” ; meticulous types may prefer to CREATE ZAPS: first and give it an appropriate type and block size, such as type T and 180.

Tony Severa
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 do
I do it? I execute CPU: 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
8088). Writing in this language is not difficult (but not always
easy), for the CPU has the following mnemonics: ADD, SUB, AND,
OR, and MOV. You'll simply have to learn them and what they do. Appendix 4 to the
ASSM manual (bound in the PTDOES manual) describes
magnetic references, Intel's excellent manual describing each
mnemonic and giving some programming examples and Lance Leben-
thal's Atlas: I Found How To Press. Whatever form you use, this
is an useful introduction.

Mnemonics aside, creating the source file requires
familiarity with the requirements of the assembler. AS5M, as
well as with EDIT or whatever program is being used to edit
the text. AS5M imposes format requirements on each line (discussed
in Section 2 of the AS5M manual) as well as on the file as a
whole; it also requires that all labels 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
AS5M manual, such as ORG, XEQ, EQU or EI.

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
PTDOES manual; suffice it to say that the program creates a new file
that mentions the desirability of using short system command
programs in the command execution buffer part of memory
beginning at CXXBF, so they won't interfere with anything else
in memory. Anything that runs by typing its name to PTDOES 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
CXXBF." At least one space before the "ORG" is required so the
assembler "will know it is an operation, not a label, see AS5M
manual 2.4. Unfortunately, we've goofed already. The
assembler doesn't know where CXXBF is; it must be told somehow
that it is at BCC9. The PTDOES system disk contains two files,
PTDOES and NTPDEFS, 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, NTPDEFS, and tell
the assembler to read the file to obtain the necessary
definitions. We do this by adding the statement "COPY
NTPDEFS" to the source file. We have skipped the COPY
statement and changed the ORG statement to "ORG BCC9" (Note that "ORG
BCC9" contains three distinct errors-- no space before ORG, BCC9
will be treated as a label and number 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 COPYPING (which will be needed
latter, as any in this case "SYST", may mean something to the reader, than to fill the source code with
numbers like BCC9 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 in memory will be the first instruction to be executed.
We must also tell the assembler the source file's name. This
should start with the first byte, which will be the one at
CXXBF. We could write "XEQ CXXBF"; it is better to substitute
the statement "XEQ?" in the PTDOES manual, ".Q", for the mnemonic of
the next byte of code. The reason is that we may someday
change the XEQ statement to load the program elsewhere. Use of the
"Q" in the XEQ statement will avoid the need to change the
XEQ statement too in order to specify that the run
command is on the same Program Microcomputer, as the loading
process; whatever form is used, this statement will be 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 08 at address A990. These require
mnemonics: one set might be: MVI A,8 <move a 8 to the A
register> and STA A990 <store the contents of the A register
at A990.> It's better practice to give the memory location,
A990, a symbolic name, let's say "atpro" for "attribute protector.
it then refer to it as such in the instruction. To do
this, we define the label by an EQU statement, "atpro EQU
A990", and rewrite the store instruction as STA atpro.
This may help us remember the significance of address A990; in
addition it will simplify changing the program if A990 turns
correct to be the wrong address, as one change in the source file
will effect the necessary substitution throughout the program
however long and however many references to the same memory
locations. Section 6.2 of the A5M manual contains one of the
instructions that everyone knows to use "XRA A" instead of "MVI
A,80"; 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 running 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 PTDOES manual. Section 7 explains the four types of
returns possible--ABTRO for serious errors, RESOP to reset
the system, RTTOP to set a trap, whatever that means, and RTTOP to
return to the normal system. We want RTTOP, the normal return.
We didn't want this to happen to us, but it didn't. If our program
is being executed in a macro through DD or SETIN, we would
want the rest of the macro (typically REATR) to be executed too.

Section 6.4.1 gives the calling sequence-- e.g., assembly
instructions are needed-- in this case "SYST", but we find on reading about RTTOP on page 7-16 that we don't have to
load any registers before the "CALL STS", 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.
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 ZAP program will be to leave our program never to return. The definitions of SYS and RETOP are in PTDOS, 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:8, Source code for ZAP, a program to defeat
- PTDOS attribute protection
- COPY PTDOS copy the definitions
- atpro EQU 06A999H the key memory location (AS81 for 1.4)
- ORG CBXUB command execution buffer
- XEG $ start at the top
- XRA A make a hold $
- 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:8.

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 uninitiated look at the 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 Halios 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 for post-mortem 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:8, sending errors to the console and omitting the cross-references. ZAP:8,ZAP:8,ZAP (<cr>). If no errors occur, ZAP is now assembled. A look at ZAP:8 reveals that the instructions are only eight bytes in all, from the AP for XOR A to the RD 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 1, block size 16H. To run it, either type "ZAP <cr>" after the PTDOS prompt or include the name ZAP in START-UP and execute it with DO or SETIN. Once ZAP is assembled and runs properly, the source file, ZAP:8, is of purely historic interest—but don't KILL it yet. Like all image files, ZAP is a string of symbols almost intelligible to humans that resists direct editing. Most of the time, making a change to ZAP requires locating the source file ZAP:8, 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 ZAP to increase contents of AS81 to 18H, 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.

Dear Tony,

I offer for publication in Proteus News "yet another" set of CP/M user routines for the SO1 as well as notes on the AS81 and a patch for the ELECTRIC PENCIL. SOLUSWEND started with the minimum routines required for the SO1 (about 75 bytes in EMBL) and driver as well as a companion disk driver and includes all of the "bells and whistles" one could desire without coming into conflict with CP/M application programs.

Following John Maguire's review of the Digital Dell N+S software (P/N 251211), 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 (200 Greenwood, Renton, WA 98055) has been working on a user-relocatable 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 read the SO1 at 8000H, followed by BASIC at 8000H, 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
Operating Policies and Procedures

January, 1988

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; foreign 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, proprietary programs can not be accepted. Remember, though, that you cannot copyright an idea, or an algorithm; 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 necessary 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 postage and handling 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. PTC BASIC internal compiled format, i.e. SAVE name
2. PTC 80BASIC internal compiled format, i.e. SAVE name,C
3. PTC 80BASIC text format, i.e. SAVE name,T
4. Assembler source files as memory images of ALS-8/80/80I 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 leading CP
   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, CR

This file format is written by the SAVE command of the newer version of SOLOUT, and by using SOLOUT/CUTER to save a memory image of an ALS-8 file or an older SOLOUT 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 SOLOUT/CUTER. Where possible, SOLOUT/CUTER should be accessed only through the entry jump table. If necessary to access any 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 distributable 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 these 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 compatible 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/CUTER block access tape format for most of our programs and data. Because of this common link between all SOL owners (and compatible 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 operating systems do not necessarily use a compatible 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 typewriter."
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 just write to the tape instead of the screen. The WRITE driver should filter out leading spaces (not leading zeros), line-feeds, nulls, and other control characters. One of the most user-friendly effects. The WRITE driver for this format was written by Richard Greenlaw to convert BASICS programs to EBCSII, and was published in Solus News, VII.3.

Similarly, the READ drivers would seek 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 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 WRITE 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 compatible with the byte access format used by EBCSII, PTOC EDIT, and the PTOC PACK and UNPACK utilities. Currently, these drivers are available to interface to the PROTEUS STANDARD BYTE ACCESS format with the following:

1. Northstar Disk TAPEOS by Joe Cumming
2. Disk copier (all disk formats) Interfaced by Bob Dobson
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 READ and WRITE drivers for any program which does I/O through SIMP and SIMP by Lewis Moseley, Jr., building on the work of Richard Greenlaw and Dave Cumming.

All of these are in the Cassette Library. The CP/M driver is on C18 with the other CP/M software. All of the others are on C9. The library welcomes comments, suggestions, and especially compatible 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 Library programs are Copyrighted. This is done solely to prevent commercial exploitation. Programs which are shown as copyrighted by the individual authors, or otherwise 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 tape 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 warranties, 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
25% Washington Ct., NE
Conyers, GA 30093

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 will be published in PROTEUS NEWS.

INTERNATIONAL NORTH STAR USERS ASSOCIATION FORMS

After considerable discussions with North Star Computer Company, a users association (similar 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 totally independent 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 coming 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:

Mr. D. A. L.
131 Highland Ave
Vacaville, Ca 95688
Larry McDavid's article on the PTDS parameter scanner (Protec News Vol 2 #5) was informative and accurate. Lately I have had some experience with the PTDS parameter scanner while working on a machine language routine to save data from a digital disk drive. I am sure where the Helios library is located and any changes since I have been there. Larry's article helped clear up the sometimes confusing description in the PTDS manual. Below is a diagram I have found useful in explaining the operation of PTDS to others.

```
A B C

*BIGPROC /1,7A@0,256:D ;NEXTPROC

Fig. 1
```

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

PTDS contains a routine called the Command Interpreter. The CI is responsible for taking the user types and passing it to the PTDS programs to be executed. The CI relies on PTDS to read the first line of what the user typed in. The PASCAL 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 which 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 PTDS. A space delimiter tells PTDS to load the file and begin executing it. Since we want to contain the program loadable in PTDS while we know how to do this correctly, assuming that our BIGPROC program has loaded and is now running, we move to part B of the command line. The user program needs to get some information from the user in order to do the terrific job we want it to do. How does it get it? Why our old friend PASCAL of course! BigPROC now makes a call to PTDS to pass us the next part of the command line and act upon the information. There are a few important things to point out about how PTDS operates on part A and B and part C, and later part D of the command line.

In part A, PASCAL is called by the CI. In part B, PASCAL is called by BIGPROC. Just how the characters returned by PASCAL will be treated upon is determined by the programmer. Go back and read Larry's article for an example of how he used the command line, 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 PTDS. When BIGPROC receives the semicolon from PASCAL it will pass control back to PTDS and CI which will now load and execute NEXTPROC. BIGPROC does that because we wrote it to do so.

In summary, PASCAL can be thought of as a vacuum cleaner with two output chutes. It starts going down the command line cleaning up the characters one by one and directing them out the chutes. Let's see how we get our chute and 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 PTDS. When BIGPROC receives the semicolon from PASCAL it will pass control back to PTDS and CI which will now load and execute NEXTPROC. BIGPROC does that because we wrote it to do so.

```
*BIGPROC /1,7A@0,256:D ;NEXTPROC
```

With due respect to Joe Maguire, I believe his assumptions regarding the demise of processor technology misses the essential point. I am well aware of the problems that P.T. has with their dealers. They have several quite well. But I do not believe this was the ultimate cause of the early death. We all regret so much.

There is no country a number of dedicated amateurs. Sometimes called hobbyists, better described as pioneers. At the front of every new technology. You find them in radio, flying aircraft, etc. They are for the most part very valuable people who are often competing with small businesses in the emerging power necessary to build and master whatever is new and challenging. In Proteus, I suspect, are largely in that group, but too few in number and so picky.

Sometime in the early part of 1978 the flow of money from amateur to hobbyist began to dry up. Fortunately for a few of us, there was second, larger, and even better suited group with a more general knowledge of hardware and software. They were not looking for fun and games and any sort of limited line. They were now looking for help and 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. None of this is new. Where would be the computing in the world without them? And how easily it would do it. In a word, application software. Turnkey is not a dirty word in this group of functional users. Everything is a work of art, and reliability is the nicest little thing you can say about anything.

A tool that makes no waves. That requires no long learning period. That does not come out with a how to use your hands and is not a hound dog. It is yet. Certainly, it is not going to pay $800 bucks for a fancy typewriter. Forget about programming anything that takes so long. Today's computer is not. Today's computer does not do any of the above. It is nothing to spend the evening and week-ends, and no programs. Understandably, here writing software for a Sanyo does not promise limited sales. Not with turnkey software and begging for anything that would run. Pat and Apple are yet to be exploited. So we wrote something that works. Something that works and most didn't give a damn about those that were here first. It was not the same.

So the cash flow slowed and creditors, burned, or chose to be replaced by the collapse of IRSI, POLY, ALTAR and the other early dropouts. We were busy with radio in advance. A common business hedge in such circumstances. PASCAL can also be something important even to such men as heads as the BRASS at P.T. That the future belonged to the likes of radio Shack, Pat and Apple. The last to have a small fortune.

Let's not fault them for saving their individual reputations by avoiding bankruptcy and lawsuits by simply shutting down. I used to think that they owed me something for supporting them in the past. Truth is, they owe it to you. They are gone. We are all quite happy. When I go into engineering now, I don't feel now that we were be exploited. We got our dollars worth in quality merchandise and in being part of something important. PASCAL can also be something important even to such men as P.T. The last to have a small fortune. We will all get the maximum out of our little computers.

As to the dealer problems that P.T. brought on itself. A consideration but no more than that. Seen an IBM, Wang, or Sperky-UVICOM store lately? The market will seek out the best. I never liked my early IBM. They are just too big, too bad as they were, and given their software development towards upgrading and adding the equipment (Microsoft, Apple, etc.) they might be the best option. I am still trying to find, and the others I can. Should need the goodwill of Buy-Shop, Computer World, or any retail store. Get your product into the hands of a few and if its solution to a problem success will beat the drums.
H-4 RECEIVES 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 disscopy the new H-4 onto it and get it mailed back to you. The reason for the $3 is to pay for the disscopy 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.

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
1.5707963 (used below)
PI=4*ATN(1)

The value of e
E=EXP(1)

To convert degrees to radians, multiply by R
R=PI/180

To convert radians to degrees, multiply by D
D=180/PI

(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))

INVERSE COSINE
C1=ATN(X/SQR(1+X*X))

INVERSE TANGENT
H1=(EXP(X)-EXP(-X))/2

H2=(EXP(X)+EXP(-X))/2

H3=(EXP(X)-EXP(-X))/2

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
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. (Somewhat this has a familiar ring to the old timers, doesn't it?)

PORTABLE PDOS
Once we get the source code, an obvious thing to do is to modify it. PDOS has the advantage over other operating systems in that it was tailored for the Sol, although it will work with any 8068/8085/8080 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 PDOS system because it is so easy to use and flexible. The major complaints are that PDOS is not ubiquitous as CP/M is, and that the Helios controller is the only one it works with. I have been in contact with a number of people who used to work on PDOS. I believe that I could put together a team to produce a new portable version of PDOS, but I need to know how many people would be interested in having the new PDOS for their system and what disks they want it to run on. What I propose to create would be a new PDOS that will have the following features:

1. It will be upward compatible from the present PDOS, with perhaps a minor change 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 or BASICS/5 programs can be transcribed automatically into PDOS files and run with little or no change. Your present copies of WordWizard, MultiMaster, etc. will run unchanged.

2. It will operate on any mixture of disks and disk controllers, such as Helios, NorthStar, Microlist, MicroMation, Thinliney, Tarbell, Morrow hard disk, NorthStar hard disk, etc. Programs will not care which disk units they are dealing with, but all disks can be incorporated into your system by simply copying the driver file from the library onto your system disk. The 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 PDOS. A CP/M program 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 VSM, and will also run with other video displays and terminals. Included in the base price will be the video editor, absolute loader, vectorized loader, disk BASIC/S, 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. PDOS 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 (the best I've ever seen), an inexpensive COROL, disk PILOT with video tape recorder control for computer aided instruction systems, and a NorthStar BASIC to EMBED conversion program. Application programs developed by you and other software 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 2.05's price. Lifeboat Associates sells CP/M for $20 to $35, depending upon the computer system. Our goal will be to match the low end, $170, but you will be getting more than CP/M for that price. The other major programs will be sold at reasonable prices.

Why bother with PDOS? 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 PDOS already. Some would like to add a hard disk (the 2-slot Helios cabinet has room for a hard disk inside). Some would like to convert their Helios to a soft-sectored double-density controller without losing the ability to run PDOS applications. And 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 whose NorthStar DOS really are missing the power of a full operating system, PTC's EMBED is similar to an enhanced version of NorthStar BASIC, so NorthStar owners with Sol's could easily move up to PDOS.

Between these two groups (Sol/Helios, and Sol/NorthStar) we could have enough interest to create the new PDOS, and it would fill a role that CP/M can't. Once we get 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 DOS, and also that PDOS has what they want. Many people who worked with both CP/M and PDOS have told me they prefer PDOS. But it needs more application programs and languages.

Considering all of this, I think it's worth trying to put together a portable PDOS. 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 PDOS (or whatever we'll call it).

MAILMASTER, WORMMAPPER, ACC Pac 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.

<table>
<thead>
<tr>
<th></th>
<th>Full Price / Manual alone</th>
<th>Minimum Purchase</th>
</tr>
</thead>
<tbody>
<tr>
<td>WordWizard</td>
<td>$300 / 35</td>
<td>$450 / 35</td>
</tr>
<tr>
<td>MultiMaster</td>
<td>$400 / 35</td>
<td>$510 / 20</td>
</tr>
<tr>
<td>MailSort</td>
<td>$520 / 35</td>
<td>$660 / 35</td>
</tr>
<tr>
<td>AccPac General Ledger</td>
<td>$600 / 35</td>
<td>$750 / 35</td>
</tr>
<tr>
<td>Financial Reporting</td>
<td>$750 / 35</td>
<td>$900 / 35</td>
</tr>
</tbody>
</table>

*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 8CG software purchase.

Stan Sokolow

Proteus

1690 Woodside Road, Suite 219

Redwood City, CA 94065
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 PICO 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 ECDBASIC.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 Pseudopos 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 RETURN 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
5076 Glendale Ct. 9E
Conyers, Ga. 30012

Thanks.

Tony Tate

END-OF-PAGE CONTROL FOR PENCIL I

I have read several commentaries on Michael Shryer Electric Pencil I. One of the main gripe's 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 2.0.

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 0B10 to 0B3F which should get you in the general area. My version looks like this:

0B10 8E OC CA CE 0A D2 AD 04 3E 0A 0D 41 09 3E 0D CD
0B20 41 09 3E 0D 0E 0D 0E 0F CD 41 09 03 26 OB 3A 7D
0B30 22 32 7E 22 2B 57 72 22 62 27 42 84 0B 0B

Change 0B29 from 41 to 38 and OB2C from 26 to 22.

0B28 CD 38 09
0B2B 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) 436-5529
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 Mill
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 PITDIS 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 if I could print out a file or program at 300 baud by entering OUT F after the PITDIS asterisk, loading BASIC, calling the program, entering a line SET OF=1 and then running the program. Then if I later wanted to use the PRINT command to transfer the PRINTER Device file from WIZARD to my PITDIS 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 =PRINTER, cfile 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 PITDIS 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 some hands on experience. First, 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 Mill
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.

Dear Tony Severa

Just read your editorial in my copy of Proteus/News for Jan/Feb, 1980 - you are a wonderful idea - 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 TD 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 Comeneco and at this writing we have the Comeneco 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 a 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. Bailey
Austin, Texas, 78747

Tony,

First, let me congratulate 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 North Star single density, dual or single drive but with an 8-108 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 @ 215-777-5818.

Ron Genova
681 Frydensbury 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. 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 PDOS. Do you know what might be causing this? Maybe PDOS 1.5 behaves differently than PDOS 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 integer argument? I wonder if TRAP could be used to call a routine to get 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 out all 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 INITPATE mystery file on Helios disk H-1. Through communications with Sasan at SLAC, I have learned that they have the INITPATE command in the PDOS STARTUP. Then, PDOS will print 'HELLO'. Normally, PDOS treats '/' the same as 'DEL'. If you want to print 'ABC/DEF', PDOS will normally give you 'ABEDE' instead. But with INITPATE, you get what you want.

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

```
TITL I/O INITIALIZER FOR NEW SLAC PDOS D01.4.1 6/15/78
COPY NYFTDUPS
ORG 0100H
XEO BEGIN
BEGIN MVI A, 0 NOP
STA 0B6CH SAP CONIN R1 WHICH SNAPS DELETES
MVI A, 7FH BACKSPACE CHAR
STA 0B5AH CONIN BACKSPACE
STA 0B5FH CONIN ECHOES FOR BS
STA 0B6FH VDM USES FOR BS
CALL SYS
DB RTTOP
END
```

Speaking of mystery files, I still don't know what the 'T' file is on Helios disk H-2. (It's DISCOPY, I got tired of typing so many characters when duplicating library disks. -Sas)

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.

A few weeks ago, I bought some of the 55 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 dealer 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 me that they tried to cooperate (Solan) 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 awful lot of potential out there for increasing membership. I hope the recent Proteus existence 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 (Dialecto) 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 cutoff 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 area of the monitor to allow for antennas 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 form 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 too 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>
acomplishes 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
#####-, BUT DOES NOT ALLOW YOU TO DELETE IT.

MAKING THIS SIMPLE PATCH TO PERMIT DELETION OF LINE #####:

EN 08C7:
80/

Ken Wong
Alberta, Canada
Arms
(or I'M stiiken with you Joe)

With each issue of PROTEUS, I feel its value more. I
look forward to finding nice liitle tidbits of information
that have been contributed by my fellow Sol alumni that I can
appreciate. I use my own system. This feels like
a level that I must now break my silence in hopes that I may
contribute something useful to you, my fellow Sollumites.

First a little feedback. I am particularly grateful 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 article "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-BLR, SD
Sales, Expano Ram = 3xk, 2 North Star disk 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, that's 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 Hoseley, Jr.
contributed an interesting programming quxicky to cut a second
or back-up tape after the original cut by hitting (cursor -
left), (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 this printer
I have to use (SE CO C900) (RETURN) (SE CO) (RETURN) (SE CO)
(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
100 POKE 51202,0,0 PRINT instead of POKE
110 POKE 51220,201
115 REM* SE C90
120 POKE 51207,0,3

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

200 POKE 51207,0

51202 is the user defined port at C802
51207 is the current output pseudo port at CB7
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 A5-4). Next find the
number (on page A5-4) in the left-hand column that
corresponds to 805 (C800) and go over the 2nd row of
numbers for decimal number representing 805, which is 2050
and add that number to 49152 to = 51202.

I found a patch in DR. DOOB'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 after the
liure, I set about to write my own patch. I hope you can use it.

For Rel. 4

<table>
<thead>
<tr>
<th>starting addr.</th>
<th>old code</th>
<th>new code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2608</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>2601</td>
<td>F0 C1 11 26</td>
<td>F0 D2 11 26</td>
</tr>
<tr>
<td>2611</td>
<td>CD D2 C7 29</td>
<td>CD ED 29</td>
</tr>
<tr>
<td>2620</td>
<td>GARBAGE 0</td>
<td>06 07 09 20</td>
</tr>
<tr>
<td>cont 2605</td>
<td>00 C2 P2 29 09</td>
<td></td>
</tr>
</tbody>
</table>

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 16K2 memory chips on the SOL mother
board with the 144K 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 Buway, 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 K-LOBAUD and DR.DOOB'S
that applies to my interests. While they are good magazines I hope
that more of the Sollumites 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
5300 Bann Circle
Dugway, UT. 84022

The "Common symptoms..." came from the
SOL Technicians manual. I plan to put out
a multi-volume set of books about SOL,
including the unpublished technical 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 SOL 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 SOME TIME IN 1980.
   IT WILL NEED ABOUT 56K MUXIMUM, 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 TERRIBLE. 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 Mybiles listing of the
   8080 assembler language version of the "Tiny Pascal" and
   am starting to try to get it running on my 51. 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 similar 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, W1 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-28, Heilon 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 1648) I found that there was apparently no way to use the printer's keyboard while running BASIC programs. Naturally, this presented an excellent opportunity 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 PTDS are replaced with two routines which check the Solos console output port location (RCB87H). If the byte is zero then the PTDS console output routine will be used. If the byte is non-zero then output will sent to the printer.

Another powerful feature of the Diablo 1648 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 movement of the print head by 1/120 of an inch, etc. The printer driver routine (starting at label PTCON) includes the necessary logic to ensure that once an escape sequence is started, an ETX will not be inserted until after 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 the use of CALL statements. 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 EBASIC it is sufficient to configure BASIC to use its internal I/O routines. Apparently BASIC uses DOS DCON and CONTEST routines for all its console input. Therefore, to use the printer keyboard, all that is necessary is to execute SKIN which will replace these 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 51287H to turn the printer on. The printer will then output until POKE 51287H.

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 SETUP command in PTDS for program listings and the like. Note however, SKIN does not support the impressive bidirectional printing of PT's 1610 driver.

Another very good use I have found for the simultaneous serial keyboard input portion of the program is in transferring 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 1200 baud. (Laiziness is the mother of invention.)

CAUTION: For reasons unknown at the present time, I am not able to run SKIN and PT's 1610, Sol2 etc. drivers at the same time. There are some strange interactions.

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. (Laiziness 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 PTDS, CPM is a dog.

Don L. Finley
345 N. RAF ALAMO, SD
36442

**
SERIAL & KEYBOARD INPUT
SERIAL I/O FOR CONSOLE OUTPUT
PTDS PORT POINTERS
NOTE: Due to code patching, following program will not run in FROM

**
ORIG RCB87H
XEQ START
START-UP INITIALIZATION
PATCH NEW ADDRESSES INTO PTDS
START LXI SP,RAMTP
LDX PYSYLD
GET SYS GLOBAL POINTER
LDX D,GLRCH
READ CHARACTER ROUTINE
DAD D
LXI D,COIN
MOVE OUR NEW CONIN ROUTINE
CALL MOVE
TO PTDS PORT LOCATION
LXI D,GLOBAL
POINT TO PT'S WRITE CHARACTER
DAD D
PUT CHAR
ROUTE POINTER
INX H
SAVE THE ADDRESS FOR LATER
MOV E,X
NOW, GET PT'S OUTPUT ROUTINE
MOV A,Y
PUT IT IN H+1
SKLD PCON+1
AND PATCH IT IN OUR CONOUT ROUTINE
PCP H
NOW PUT ADDRESS OF OUR ROUTINE IN PT'S
LXI D,1648
POINT TO NEW CONOUT ROUTINE
CALL MOVE
MOVE IT TO PT'S ROUTINE POINTER
LXI D,GLRCH
POINT TO PT'S CONSOLE TEST FOR WAITING

CONT'd. 8/14.
**CHARACTER ROUTINE**

```
DAD D
LXI D, CONTS
CALL MOVE
LDL SYSGL0
LXI D, GLRIO
DAD D
ORHL #99
SRL X+1
STA X+1
LDA BFLAG
ROR
BRA
JMP 8BCB0H
ALL DONE
```

* ROUTINE FOR MOVING ADDRESS IN D,E TO MEMORY

```
MOVE
INX H
MOV M,D
RET
```

* THE ROUTINE CONTS WILL BE USED IN THE PLACE OF

```
* THE PTDO routine CONTST

CONTS IN KSTAT TEST KEYBOARD STATUS
CMA ROLL IT OVER
ANT 81
RNL SOMETHING THERE IF <>0
IN SSTAT TEST SERIAL STATUS
ANT 4BH
NOTHING THERE EITHER
IN SDATA SOMETHING THERE, BUT WHAT IS IT
ANT 7FH
CPI ACK IS IT ACKNOWLEDGED?
JZ X1 IF SO THEN CLEAR ACK FLAG
STA CHRB NOPE, SAVE IN CHARACTER BUFFER
MOV A,1
STA BFLAG SET CHARACTER-IN-BUFFER FLAG
ORA A SET CHARACTER WAITING FLAG
RET GO HOME
```

* THE ROUTINE CONIN IS TO APPEAR THE SAME AS PTDO

```
* ROUTINE CONIN

CONIN LDA BFLAG TEST CHAR.-IN-BUFFER FLAG
ORA A
JZ X2 NO CHARACTER
XRA A
STA BFLAG SET CHAR.-IN-BUFFER FLAG
LDA CHRB GET CHAR. FROM BUFFER
RET TAKE IT BACK
```

```
X2 IN KSTAT CHECK KBORD STATUS
CMA
ANT 81
JNZ X3 SOMETHING ON KEYBOARD
IN SSTAT CHECK SERIAL STATUS
ANT 4BH TEST IF RECEIVER FULL
JZ X2 IF NOT, KEEP LOOKING
IN SDATA GET WHAT IT IS
ANT 7FH
CPI ACK IS IT ACKNOWLEDGED?
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 GLB10 CHECK IF BINARY FLAG
ORA A
JZ X5 WANTS PARITY STRIPPED
POP PSW
NOPE
RET
```

X5 POP PSW YEP, STRIP PARITY
ANA 7FH
RET

* 1640 DRIVER ROUTINE
* NOTE: DOES NOT SUPPORT GAUDY BI-DIRECTIONAL PRINTING

```
A1640 MOV B,A SEE WHERE OUTPUT GOES
LDA OFLAG ORA A, B IF NON-ZERO THEN GOES TO PRINTER
MOV A,B
IF ZERO THEN TO PTDO
PTCON JZ 0 OUTPUT TO CONSOLE, ADDRESS PATCHED
LDA AFLAG TEST IF NEED TO WAIT ON ACK
ORA A
CNZ WAIT
B1640 IN SSTAT GET SERIAL STATUS
RAL
JNC B1640 WAIT TILL XMITTER EMPTY
LDA AFLAG SEE IF ESCAPE SEQUENCE SET
ORA A
MOV A,B GET CHARACTER BACK
JNZ ESC01 IF SO THEN SPECIAL HAND.
CPI ESC
JNZ OUTIT IS IT ESC.
JNZ OUTIT IF NOT, THEN OUTPUT IT
ESCSE OUT SDATA OUTPUT ESCAPE
MOV A,1
STA EFLAG SET ESCAPE SEQUENCE FLAG
LDA NHCHAR
INC A
STA NHCHAR
MOV A,B
RET
```

* SPECIAL HANDLING FOR CHARS FOLLOWING ESCAPE

```
ESC01 CPI 99 HT REQUIRES ANOTHER CHR.
CPI 88h
JZ ESC05 VT ALSO
CPI 8Ch
JZ ESC05 FF ALSO
CPI 8Dh
JZ ESC05 CR ALSO
CPI 16h
JZ ESC05 SYN ALSO
CPI 16h
JZ ESC05 RS ALSO
CPI 16h
JZ ESC05 US ALSO
OUT SDATA SINGLE CHAR AFTER ESC
XRA A
STA EFLAG CAN TURN OFF ESC FLAG.
LDA NHCHAR
INC A
STA NHCHAR
MOV A,B
GET CHAR BACK
RET
GO BACK
ESCSE OUT SDATA ANOTHER ESC SEQ. CHAR COMING
MOV A,B
LEAVE ESC. FLAG SET
RET
```

```
OUTIT OUT SDATA PRINT IT OUT
LDA NHCHAR TEST IF NEED TO INSERT AN ETX
INC A
STA NHCHAR
CPI 101 PUT IN AFTER 10H CHARACTERS
JNC 1640 IF LESS THAN
MV1 A,1
JNC 1640
STA AFLAG SET ACK FLAG
STA NHCHAR
JNC 1640
LEAVE CHARACTER COUNT
C1640 IN SSTAT OUTPUT ETX CHARACTER
RAL
JNC C1640
MV1 A,ETX
OUT SDATA
```

Cont. B.15.
Dear Stan,

I'm glad that I found out about your Newsletter, especially since PCT 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; it didn't have a Theory Of Operation section. We received our Helios II in Sept. 1977 and had only received notices up to #4.

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

- IMSAI mainframe
- TDL 2-80 processor
- PTC 160-1, CUTS board, and Helios II computer/speechlab
- Heathkit H14 printer
- Addmater paper tape reader
- Software Tech. Music System
- Objective Design Programmable Character Generator
- Crossmo electronic
- MTS 88-4 Parallel I/O board
- IMSAI S10 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 ZZPPELE monitors. I used the command buffer and cassette tape features from CUTER, and the extensive command set from the ZZPPELE monitor (it has 26 commands) with the entry vectors for both monitors. It occupies a little over 4K, and the assembler 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 50 characters/line and the other at 122 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 page edge is punched. I use the formatted driver for BASIC listins, if you are interested. I can send PROTEUS source listings of these 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 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 set 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 out of space and tries to write all over the system disk (which has happened to us twice).

We are using an Mtek 32K memory board with 16K of ram on it for PTDOS. We've had no DPM problems with this board, but it runs very hot. We're planning to replace it with a lower powered board in the future. With almost a full computer (the IMSAI holds 22 boards), a lot of heat is generated as is.
To take speculation several steps further my ideal home micro system would include the hardware just listed plus a real time clock/vectored interrupt board and a software System Executive that would time-share foreground/background programs according to priorities requested through the software.

With such a system 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 interrupt other activities it is doubtful 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 acquisition 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 MEGA BETA-1 TAPE SYSTEM AND THE SOL

The Beta-1 Tape System, available from MEGA (7026 O.W.S. Road, Yucca Valley, CA 92284) is a digital tape mass storage system that is suitable for use with any 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 drives, 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 CUPS 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 MEGA 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 Beta-1 controller contains an 8035 microprocessor with a 1K byte EPROM program that accepts a quite a few ASCII-encoded commands. Of course, it's not all that simple, either. The SOL has its own 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.
The diagram on the next page (Figure 1) shows the connections to be made between beta-l lines and SOL parallel port lines. In order to make the connections, you first need to remove the conductor from the Beta-l cable, and you need to get a 25-conductor cable with a 25-250 connector connector at a end for the SOL side. You must then separate the individual wires in each cable, and strip the last inch or so from each end. New, 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 no choice. (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. (After a week of hard work I had to learn how to carefully separate and strip those 25 individual wires) The big problem is that directly connecting wire to wire quickly yields a humble, 0, and you can't even detect the short. In order to make any sense of what is going on, you will need to connect the wrong lines along the way. I am careful, at least in checking my work for mistakes, and I discovered quite a few of the right way. Then I hit upon the idea of using 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 twisted wires, and used two wires of copper wire ends forming reasonable places to connect the Beta-l and SOL cables. I then twisted together the corresponding wires, and finally checked for proper connection by testing from the Beta-l connector end to the corresponding pin in the SOL connector with an ohmmeter. Once I assured myself that the wires were correctly connected, I recommended the Beta-l 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-l respond with its "ready" signal (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-l 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.
3. Built a perfboard with two "connection strips" consisting of the ends of short telephone-type wires, which were inserted crosswise in the appropriate order, so the cables 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-l connector and connected the SOL end.

There are more additional steps involved:

1. The SOL's "data acknowledge" input control line requires the corresponding Beta-l signal to be inverted. I did this by using a 7406 IC on the perfboard, connecting the Beta-l signal to pin 1, the SOL line to pin 2, and the +5V and ground lines to the Beta-l lines of the same function. I added a 470-ohm pull-up 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 and 7406.
2. In order for the Beta-l to supply the +5V needed by (1), you must install a jumper (+39) on the beta-l board, and follow the disassembly/assembly instructions supplied with the unit.
3. Three Beta-l lines and four SOL lines are left unconnected, and have no "Beta-l" ground lines and the SOL's single ground are connected together.

Now for some options:

Using the Beta-l with the SOL means having to write driver routines. I considered writing replacements for the CUPS routines in SOLOS, but that would be too much work for starters, so I'm writing routines that allow me to manipulate beta-l files directly, connecting the assembler directly to a Beta-l tape file. The Beta-l controller program accepts commands like: open file, kill file, list directory, read, seek, write, and revd. 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-l, however, that seems to require some operating system level program design.

The Beta-l 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 the 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 CUPS 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-l provides mass storage capacity comparable to a single-sided double density floppy, at a lower price (plus lower speed); it is an improvement of considerable magnitude over audio recording. The fact that an S-100 board is needed to interface the Beta-l makes it attractive for the SOL, especially since the SOL interfaces are reasonable to connect. Software is a problem, but if enough people acquire Beta-l's, software should become available.

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

---

**Figure 1**

**Beta-l Signal #** | **Direction** | **SOL Parallel Port Pin #** | **Signal**
--- | --- | --- | ---
1 | => | Note A | +SV
2 | => | 25 | Input data 0
3 | => | 24 | Input data 1
4 | => | 23 | Input data 2
5 | => | 22 | Input data 3
6 | => | 21 | Input data 4
7 | => | Note Br 16 | Data acknowledged
8 | => | 20 | Input data 5
9 | => | 19 | Input data 6
10 | => | 18 | Input data 7
11 | => | 17 | Output data 0
12 | => | 16 | Output data 1
13 | => | 15 | Output data 2
14 | => | 14 | Output data 3
15 | => | 13 | Output data 4
16 | => | 12 | Output data ready
17 | => | 11 | Output data 5
18 | => | 10 | Output data 6
19 | => | 9 | Output data 7
20 | => | 8 | Output data 8
21 | => | 7 | Output data 9
22 | => | 6 | Output data 10
23 | => | 5 | Output data 11
24 | => | 4 | Output data 12
25 | => | 3 | Output data 13
26 | => | 2 | Output data 14
27 | => | 1 | Output data 15
28 | => | 0 | Output data 16
29 | => | 19,18,23,27,31 | ---
30 | => | 2 | Ground
31 | => | 1,3,14,15 | (Unconnected)

Notes:

(A) with beta-l jumper W30 installed; supplies +5V for (B)
(B) Beta-l signal 11 is inverted by circuit below before going to SOL pin 16
(C) It is understood that some SOL parallel ports have their own in different positions from the one depending upon the revision level. Mine is Rev. E SOL. Check yours before starting!

---

*Note: The portable PDP10 I talk about on page 15 could be interfaced to a block-addressable tape like the Mera. 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.*
I have been in 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 reasonable 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 Severs Data Service
131 Highland Ave
Vacaville, CA 95688

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 32K ERA can be expanded to 64K.........$350.00
Fully wired and operational keyboards.................................$350.00
3.5" B-boards (includes PROMS).............................$150.00
(Or will sell individual items to highest bidder.)

Highest bid
Whole Subsystem 'B' in IMSAI frame......................$1000.00

103801. Rev B Backplane boards......................$7.50
107801. 2708 Personality modules......................$7.50
110211. Regulator.................................$7.50
110221. Rev C Backplane boards......................$7.50
301804. Rev C Sol main logic board......................$50.00
301804. Rev C Helios Format board......................$50.00
301804. Rev D Helios Format board......................$50.00
382001. Helios Regulator board......................$50.00

Sol Board Set (103801, 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
787-762-9514
(Can also be utilized by Tony's Data Service)

Ray Pohl
3388 Moraga Street
Lafayette, CA 94549
415-283-6630
(Can also be utilized by Tony's Data Service)
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 V3.2 (see Dr. Posch'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 "FM", and Lee has provided a hardware modification which relocates the entire on-board RAM/ROM/VRM 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 EU's effective size.

Lee had announced his intention to attend the Dec. 4th PROTONS 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.

Subject: CP/M for SOL on North Star ED

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 D1B 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 2K distribution version (BB1D or BA1D in my 4KX version) is placed at byte 5B when a cold boot is done and therefore this byte controls the 10 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 therefore insert 5A4H at the above location, do a warm boot and then do the SAVEUSER routine. Each time the system is cold booted after this, byte 5B is 6A 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 6C18H in the 4KX version.

For any SOL owner who has access to a 24 x 60 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 different style of operation after being used to PTOS.

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

J.G. Zeratsky

Dear Tony,

1. I just got an IDS-448 "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 port and then everything goes to the printer not very 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-288 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, 16 hard sectors, 512 bytes/sector and 40 traks for a capacity of 204800 bytes.

Bob Freeman
1857 Ouray Ave
Grand Junction, Co 81501
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
<th>Article</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EDITORIAL</td>
</tr>
<tr>
<td>2</td>
<td>Introduction to ASSM - The PTDOO Assembler</td>
</tr>
<tr>
<td>3</td>
<td>LETTER: Electric Pencil and North Star</td>
</tr>
<tr>
<td>4</td>
<td>Cassette Library Operating Procedures</td>
</tr>
<tr>
<td>5</td>
<td>Formation of the International North Star Users Ass.</td>
</tr>
<tr>
<td>6</td>
<td>PTDOO Scanner (PSCAN) Notes</td>
</tr>
<tr>
<td>7</td>
<td>The Late Lamented (More on P.T.'s demise)</td>
</tr>
<tr>
<td>8</td>
<td>5th West Coast computer Faire Notes</td>
</tr>
<tr>
<td>9</td>
<td>NOVICE CORNER: Derived Functions</td>
</tr>
<tr>
<td>10</td>
<td>NEWS FROM PROTEUS: Source Code Progress Report, Portable PTDOO, Helios Software Still Available</td>
</tr>
<tr>
<td>11</td>
<td>Software Wanted</td>
</tr>
<tr>
<td>12</td>
<td>North Star Double Density I/O Routines</td>
</tr>
<tr>
<td>13</td>
<td>LETTER: Novice Learns How To Print in PTDOO</td>
</tr>
<tr>
<td>14</td>
<td>LETTER: Need for Novice Oriented Articles</td>
</tr>
<tr>
<td>15</td>
<td>LETTER: Reading PA Computer Club</td>
</tr>
<tr>
<td>16</td>
<td>LETTER: PASCAL</td>
</tr>
<tr>
<td>17</td>
<td>Programming Quickie/Software Technology Music System</td>
</tr>
<tr>
<td>18</td>
<td>LETTER: ARMS and North Star</td>
</tr>
<tr>
<td>19</td>
<td>LETTER: PASCAL / Alg-8</td>
</tr>
<tr>
<td>20</td>
<td>LETTER: Serial Keyboard IN</td>
</tr>
<tr>
<td>21</td>
<td>LETTER: Helios Write Protect Modification</td>
</tr>
<tr>
<td>22</td>
<td>LETTER: Combined Tape and Disk Storage</td>
</tr>
<tr>
<td>23</td>
<td>MECA BETA-1 Tape System</td>
</tr>
<tr>
<td>24</td>
<td>ANNOUNCEMENTS</td>
</tr>
</tbody>
</table>

## PROTEUS NEWS

Formerly Called Solus

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.

Copyright (C) 1979 PROTEUS. All rights reserved.

Permission is hereby granted to reproduce computer programs contained herein, provided that PROTEUS and the author are given credit.

Instructions to contributors: Letters and articles may be submitted in camera-ready form or on Solt/Kuts cassettes or Helios (PTDOO) 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, Cuts, PTDOO input routines. Mdia 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
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 let's 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.
**GENERAL OVERVIEW OF INPUT ROUTINES**

- **SINP**: C61F: Pseudopart number recovered from IPRT (C866); Character returned in "A": "I" flag set if no character found. Only the "A" register affected. No stripping is done of the MB (Most Significant Bit). If a character was found, status is reset. Default device is keyboard. SET and certain errors will cause the default condition.

- **AINP**: C622: Uses pseudodevice specified by "A" register.

**KSTAT**: CR2: SOL KEYBOARD
(Pseudodevice)

**SSTAT**: CR4: SOL SERIAL PORT
(Pseudodevice)

**FSTAT**: CR6: SOL PARALLEL PORT
(Pseudodevice)

**EXRT**: C63B: USER-DEFINED INPUT "DEVICE"
(Pseudodevice)

**GETS**: address from UPRT at C868-C869.

**CHARACTERISTICS COMMON TO ALL INPUT ROUTINES**

Character returned in "A": "I" flag set if no character found; only the "A" register affected. No stripping is done of the MB (Most Significant Bit). Status is reset.

**TYPICAL CALLING PROGRAM**

- **PROG**: CALL INPUT
  - check for a character.

- **JNZ**: PROC
  - if none available, wait.

- **ANI**: 7F
  - 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 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.
- Store it in TEMPorary location.
- *** A 0# code is stored if no character ***
- *** "I" 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.

**GOTIT**

- LDA TEMP
  - check TEMPorary storage first.
- ORA
  - If empty, set "I" flag.
- JNZ GOTIT
  - If OK, skip loop.

**LOOP**

- CALL INPUT
- JNZ LOOP
  - Keep trying.

**GOTIT**

- ANI 7F
  - Strip off MSB.

**PUSH PSW**

**XRA A**

**STA TEMP**

**PUS PSW**

**XRA A**

*** character is now in "A" ***

**TEMP contains 0# ***

**THE MAJOR ROUTINES***

**THE SINP ROUTINE**

**SINP**: C61F: Pseudopart number recovered from IPRT (C866); Character returned in "A": "I" 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 C61F, and returns the character in the "A" register. If no character was ready, then the "I" flag is set upon return. The routine preserves all registers except "A", in which the character is returned. If a character was found, 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 the Keyboard, Serial Port, Parallel Port, and User-Defined Input Routine. Which of these pseudodevices will be used depends upon the contents of a status byte called IPRT, located at C866. The binary value from 0-3 found at this location will be loaded into the "A" register, and then control is passed to the input routine (which is discussed next). The default value at IPRT is 0, for the SOL keyboard.

The value at IPRT may be changed in two ways. From the keyboard, when in the SOL COMMAND MODE, typing 50 I X (where X is a value from 0-3), will cause X to be the current pseudodevice. It should be noted that a RESET will cause the default value of 0 to be loaded again.

The second way to load a value into IPRT is under program control. For instance, the series of instructions MOV A,9 then STA C866 would cause the User-Defined Input Routine to become current. By the same token, a program can find out what the current pseudodevice is by using the instruction LINI C866.

**THE AINP ROUTINE**

**AINP**: C622: Enter with pseudodevice # in "A": Character returned in "A": "I" 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.

**AINP** stands for "A-INPUT." The AINP routine is accessed by placing a pseudodevice code in the "A" register and calling C622. The pseudodevice can have the following values:

- 0 = KSTAT: CR2, the SOL KEYBOARD
- 1 = SSTAT: CR4, the SOL SERIAL PORT
- 2 = FSTAT: CR6, the SOL PARALLEL PORT
- 3 = EXRT: C63B, the SOL USER-DEFINED ROUTINE

(It vectors output to the address found at UPRT)

When the AINP routine is called, the value in the "A" register is used to vector to one of the above mentioned pseudodevices. It is up to the user's program to insure that the "A" register contains a pseudodevice number from 0-3. (The AINP program will discard all but the two least significant bits, as it contains its own A3 instruction).

**PSEUDODEVICES***

**BESIDES USING SINP AND AINP, PSEUDODEVICES MAY BE DIRECTLY CALLED**

**BY KEYBOARD**: KSTAT: CR2

Character returned in "A": "I" 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 a pseudodevice 0, the default pseudodevice. Besides being available via calls to SINP and AINP, it may be directly accessed via a CALL KSTAT (CALL CR2). It is suggested that either SINP or AINP (with "A" set to 0) 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.
UNDERSTANDING AND USING YOUR SOL...CONTINUED

KEYBOARD PORT ASSIGNMENTS:

Keyboard Status Port=START=FA
Status Bit=|KDB|bit 0 (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

KESTAT IN START ;GET STATUS WORD.
CMA ;INVERT IT FOR PROPER RETURN.
ANI KDB ;TEST ONLY KEYBOARD BIT.
RZ ;RETURN WITH "Z" SET IF EMPTY.

IN |KDATA ;GET CHARACTER.
RET ;*** UNSTRIPPED CHARACTER IN "A" ***

1: SERIAL PORT: SSTAT= C042
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 a parallel port. Besides being available via calls to SINP and AINP, it may be directly accessed via a CALL SSTAT (CALL C042). 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 7 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 8 connected (with the appropriate switchers 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) |
| 9 | CLO | CURRENT LOOP OUT |
| 10 | LRI | LOOP RECEIVER 1 |
| 11 | LRI | LOOP RECEIVER 2 |
| 12 | DTR | DATA TERMINAL READY (OUT) |
| 13 | LCS | LOOP CURRENT SOURCE |

The following port information is made available for those who might find it useful.
Serial Status Port=|GERS|=F6
Status Bit=|GDB| 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.

<table>
<thead>
<tr>
<th>BIT</th>
<th>NAME</th>
<th>FUNCTION</th>
<th>ACTIVE DIRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SCO</td>
<td>Serial Carrier Detect (EIA)</td>
<td>1=Carrier on</td>
</tr>
<tr>
<td>1</td>
<td>SDR</td>
<td>Serial Data Set Ready (EIA)</td>
<td>0=Link OK</td>
</tr>
<tr>
<td>2</td>
<td>SPE</td>
<td>Serial Parity Error (UART)</td>
<td>1=Error</td>
</tr>
<tr>
<td>3</td>
<td>SPE</td>
<td>Serial Framing Error (UART)</td>
<td>1=Error</td>
</tr>
<tr>
<td>4</td>
<td>SOE</td>
<td>Serial Overrun Error (UART)</td>
<td>1=Error</td>
</tr>
<tr>
<td>5</td>
<td>SCTS</td>
<td>Serial Clear To Send (EIA)</td>
<td>0=Clear</td>
</tr>
<tr>
<td>6</td>
<td>SD</td>
<td>Serial Data Ready (UART)</td>
<td>1=Ready</td>
</tr>
<tr>
<td>7</td>
<td>STBE</td>
<td>Serial Xmit Buff Empty (UART)</td>
<td>1=Empty</td>
</tr>
</tbody>
</table>

The SOL SERIAL INPUT ROUTINE follows:

SSTAT IN |SSTAT |GET SERIAL STATUS WORD.
ANI |SDB |GET 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 a parallel port. 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=|START|=FA
Status Bit=|BDR|bit 1 (LOW means READY)
Parallel Data Port=|DATA|=F0

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 | INPUT ENABLER | (IN) 1=ENABLED |
| 4 | DR | *DATA READY | (IN) 0=READY |
| 5 | *IAK | *INPUT ACKNOWLEDGE | (OUT) 0=AACKNOWLEDGE |
| 6 | ID | INPUT DATA, BIT 7 | (IN) |
| 7 | ID | INPUT DATA, BIT 6 | (IN) |
| 8 | ID | INPUT DATA, BIT 5 | (IN) |
| 9 | ID | INPUT DATA, BIT 4 | (IN) |
| 10 | ID | INPUT DATA, BIT 3 | (IN) |
| 11 | ID | INPUT DATA, BIT 2 | (IN) |
| 12 | ID | INPUT DATA, BIT 1 | (IN) |
| 13 | ID | INPUT DATA, BIT 0 | (IN) |

CONTINUED ON PAGE 4.
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 runtime pseudoprint to # 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 AINF. 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 your own particular needs, but believe me, the ability to access such programs using SINP and AINF is a powerful feature.

This documentation provided courtesy of
Fr. Thomas McGahes
Don Bosco Tech
292 Union Ave.
Paterdon, N.J. 07580 phone (201) 595-8800

LOADING AND SAVING MICROFOLIS SOURCE/TEXT FILES
ON SOLIDS Cassettes --- VERSION 2.1
By Melvin M. Dalton

INTRODUCTION
This program provided for SOL-20 users who also have one or more MICROFOLIS Mod I/II drives. The program assumes you are using MICROFOLIS POS VS 4.0 (1979). The assembly language program resides in the MBDOS Application Program area starting at 2800H. It is designed to be called as an implicit command from MBDOS.

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

This program operates only for MICROFOLIS users.

The programs published in PROTEXT 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. PROTEXT Library byte mode format is:
   <byte line number> <20H> <text> <CR>
2. ALSB/SCS16 block mode format is:
   <line length> <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. ALSB/ESCG16 block mode files MUST be exactly as shown above except the space after the line
      number may be omitted 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
   2. Boot MMDS and SAVE 'CASSETTE' 2800 2611 1B
   3. CASSETTE is now an implicit MMDS command.
   B. Slow but sure:
   1. Boot MMDS and ENTER LINEEDIT
   2. Enter source text given below (watch those 
      <CR>'s!!!).
   3. Be sure you have source file called SOLEGU,
      if not, copy from below and SAVE 'SOLEGU'.
   4. Assemble program, add any if-then's and
      execute a TYPE 'CASSETTE' 1B command.
   5. CASSETTE is now an implicit MMDS command.

SYNTAX OF IMPLICIT COMMAND CASSETTE

CASSSETTE "[drive #1]<diskfile name>" "<SDL file name>" "<option>"
[drive #1] must be 0-3 (default is 0)
"<SDL file name>" may not exceed 5 characters
option "RA" means read disk & write ASCII format tape
option "BB" means read disk & write byte mode tape
option "BA" means read ASCII format tape & write on disk
option "BB" means read byte mode tape & write on disk

EXAMPLES:
   CASSETTE 11: CASSETTE 1" WB" --- this command will
   create an assembly source file called "CASSETTE" on
   drive 1 from the byte-mode source/ASCII file
   called CASSET1.
   CASSETTE 'TEST' 'MMD' 'RA' --- this command will make
   an ASCII format recording called MMD of the MMDS
   type 4-7 source/ASCII file found on drive 0 and
   called 'TEST'.

The program will sign on with the message:
CASSETTE UTILITY --- MMDS/SOLODS VERSION 2.1 11/79

ERROR MESSAGES
All MMDS 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 explanation.
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 'SOLEGU'

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>NLIST</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>IFORT</td>
<td>EQU OC0304H</td>
</tr>
<tr>
<td>0020</td>
<td>IFORT</td>
<td>EQU OC0074H</td>
</tr>
<tr>
<td>0030</td>
<td>UIPORT</td>
<td>EQU OC0030H</td>
</tr>
<tr>
<td>0040</td>
<td>UIPORT</td>
<td>EQU OC0028H</td>
</tr>
<tr>
<td>0060</td>
<td>USARE</td>
<td>EQU OCAB4H</td>
</tr>
<tr>
<td>0060</td>
<td>FCBA11</td>
<td>EQU OC0515H</td>
</tr>
<tr>
<td>0070</td>
<td>FCBA12</td>
<td>EQU OC0516H</td>
</tr>
<tr>
<td>0080</td>
<td>FUUF11</td>
<td>EQU OC0583H</td>
</tr>
<tr>
<td>0090</td>
<td>FUUF12</td>
<td>EQU OC0584H</td>
</tr>
<tr>
<td>0100</td>
<td>SCREEM</td>
<td>EQU OC0900H</td>
</tr>
<tr>
<td>0110</td>
<td>DLL</td>
<td>EQU OC29CH</td>
</tr>
<tr>
<td>0120</td>
<td>START</td>
<td>EQU OC0018H</td>
</tr>
<tr>
<td>0130</td>
<td>INIT</td>
<td>EQU OC0019H</td>
</tr>
<tr>
<td>0140</td>
<td>RETRN</td>
<td>EQU OC0041H</td>
</tr>
<tr>
<td>0150</td>
<td>FOPEN</td>
<td>EQU OC0078H</td>
</tr>
<tr>
<td>0160</td>
<td>FDCLS</td>
<td>EQU OC004AH</td>
</tr>
<tr>
<td>0170</td>
<td>RDBYT</td>
<td>EQU OC004DH</td>
</tr>
<tr>
<td>0180</td>
<td>WRBT</td>
<td>EQU OC0109H</td>
</tr>
<tr>
<td>0190</td>
<td>KTBLK</td>
<td>EQU OC0133H</td>
</tr>
<tr>
<td>0200</td>
<td>WRITL</td>
<td>EQU OC0169H</td>
</tr>
<tr>
<td>0210</td>
<td>SOUT</td>
<td>EQU OC0191H</td>
</tr>
<tr>
<td>0220</td>
<td>ADUT</td>
<td>EQU OC01C4H</td>
</tr>
<tr>
<td>0230</td>
<td>SINF</td>
<td>EQU OC01F1H</td>
</tr>
<tr>
<td>0240</td>
<td>ANF</td>
<td>EQU OC0229H</td>
</tr>
<tr>
<td>0250</td>
<td>ADUT</td>
<td>EQU OC02DEH</td>
</tr>
<tr>
<td>0260</td>
<td>THEAD</td>
<td>EQU OC081CH</td>
</tr>
<tr>
<td>0270</td>
<td>DHEAD</td>
<td>EQU OC0B2DH</td>
</tr>
<tr>
<td>280</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* * * *

0000 | NLIST |
0010 | IFORT | EQU OC0304H |
0020 | IFORT | EQU OC0074H |
0030 | UIPORT | EQU OC0030H |
0040 | UIPORT | EQU OC0028H |
0060 | USARE | EQU OCAB4H |
0060 | FCBA11 | EQU OC0515H |
0070 | FCBA12 | EQU OC0516H |
0080 | FUUF11 | EQU OC0583H |
0090 | FUUF12 | EQU OC0584H |
0100 | SCREEM | EQU OC0900H |
0110 | DLL | EQU OC29CH |
0120 | START | EQU OC0018H |
0130 | INIT | EQU OC0019H |
0140 | RETRN | EQU OC0041H |
0150 | FOPEN | EQU OC0078H |
0160 | FDCLS | EQU OC004AH |
0170 | RDBYT | EQU OC004DH |
0180 | WRBT | EQU OC0109H |
0190 | KTBLK | EQU OC0133H |
0200 | WRITL | EQU OC0169H |
0210 | SOUT | EQU OC0191H |
0220 | ADUT | EQU OC01C4H |
0230 | SINF | EQU OC01F1H |
0240 | ANF | EQU OC0229H |
0250 | ADUT | EQU OC02DEH |
0260 | THEAD | EQU OC081CH |
0270 | DHEAD | EQU OC0B2DH |

* * * *

# PROGRAM TO TRANSFER CUTS BYE
# MODE FILES TO/FROM MIRCOPOLIS
# SOURCE/TEXT FILES.
# WRITTEN BY MELVIN M. DALTON
# VERSION 2.1 NOVEMBER 1979
# TAB 10:15:27
# LINK 'SYS01'
# LINK 'SYS02'
# LINK 'SOLEGU.ORG REPORT'.

# PARSE COMMAND LINE WITH FOLLOWING FORMAT:
# CASSETTE "<drive #1><diskfile name>"
# "<SDL file name>" "<option>"
# "<drive #1>" MUST BE 0-3 (DEFAULT IS 0)
# "<SDL file name>" MAY NOT EXCEED FIVE CHAR.
# OPTION "BB" = READ DISK & WRITE ASCII FORMAT TAP.
# OPTION "WB" = WRITE ON DISK FROM BYTE MODE TAP
# OPTION "RA" = READ ASCII FORMAT TAP
# OPTION "WA" = WRITE ON DISK FROM ASCII FORMAT TAP

MICROPOLIS...CONTINUED. 46
CONTINUED

2C11 FE 04 CPI 4 IFS IT TYPE 4?
2C13 FE 11 MVI A,17 WRONG FILE TYPE
2C15 2BF 1C JNZ @DISKERRO MESSAGE OUT
2C18 C9 RET

2C19 E5 PREPARE TAPE FOR BYTE MODE

2C22 2F 2C BYTEOPEN PUSH H SAVE HDR PNTR ON STACK
2C23 3E 3C CALL TRANSFER MOVE THE NAME
2C24 3E C1 MVI A,'A'+$0H ISOLATE TAPE
2C25 32 26 C8 STA THEAD$ CONSISTENT FOR STACK
2C26 21 28 2C LHI H,$9 ERROR IN SOLDS 1,3
2C27 22 25 2F XTHL IPTR TO HEADER
2C28 22 33 2F MVI A,$1 ISTATE TAPE FILE 1
2C29 22 0D 2F CALL OPENERR OPEN FILE
2C2A DA 66 2D JC OPENERR FOPEN FILE
2C2B E1 POP H IF NO ERROR RETURN
2C2F C9 RET

2C30 E5 MAIN TRANSFER LOOP

2C33 DA 01 READLOOP MVI B,1 IPTR TO FILE 1
2C34 DA 7A 2B CALL @FINXPOS ITHROW FIRST BYTE OUT
2C35 DA 93 2D JC EXIT2 IFS LENGTH 01?
2C38 00 2D BCR C
2C39 CA 59 2D JZ DISKEOFF IEND OF FILE
2C3C 21 A0 01 MVI $D+BUF LENGTH IN
2C3F 3E 3F MVI $E+buf J'DE'
2C40 CD C2 1B CALL @LOADDATA GET LINE
2C41 DA 33 2D JC EXIT2 TAPE FILE 1
2C42 46 2F WRITETAPE MOV B,M GET BYTE
2C43 3E 01 MVI A,$1 TAPE FILE 1
2C44 E5 PUSH H
2C45 D0 PUSH D
2C46 CD 10 2D CALL WRBWT INPUT BYTE ON TAPE
2C48 D0 POP D
2C49 E1 POP H
2C4A DA 70 2D JC WRITERR
2C4B 23 INX H IPTR NEXT BYTE
2C4C 1B DCR E ICOUNT LENGTH
2C4D 2B 2C JNZ WRITETAPE INRECORD LINE
2C4E 3A 30 2C JMP READLOOP INNEXT LINE

2C50 E5 ROUTINE TO MOVE SOL

2C53 E5 FILE NAME FROM COMMAND
2C55 E5 LINE TO DESIRED HEADER.

2C59 E5 FILL WITH 00H TO MAKE

2C5B E5 FIVE CHARACTERS OR OUTPUT

2C5D E5 ERROR MESSAGE IF NAME

2C5F E5 TO LONG

2C60 19 29 TRANSFER LHI D+ASCIIBUF1 TAPE NAME
2C62 0E 05 MVI C+$ ISOL NAMES <=5
2C63 5E F0 TRANSFER1 DAX D
2C64 5D 2C CPI 20H IFS IT 'SPACE'
2C65 2C 49 XRA A JNZ TRANSFER2 IF 50 CHANGE TO 0
2C67 2F 2F TRANSFER2 MOV M,A INPUT IN HEADER
2C69 77 INX H
2C6A 13 INX D
2C6C 00 DCR C INX MAX CH.

2C6F 22 2C INX TRANSFER1 INEX STREET
2C70 1A LDX D ISOL CHARACT.
2C71 FE 20 CPI 20H IFS IT 'SPACE'
2C72 77 XRA A INDX ERROR
2C74 AF SYMERR XRA A IFS SYNTAX ERROR
2C75 3B 81 1C JMP @DISKERRO

2C78 E5 TEST IF CHARACTER IS NUMBER

2C7E 0C NUMBCRN CP 40H INOT A NUMBER

2C83 E0 FP CPI 30H REVERSE CARRY
2C87 3D BF DB RC
2C8F C9 RET

2C90 E0 PREPARE TO WRITE ON DISK

2C94 0B WRITEDIK MVI C+0 IPTR TO ASCIIBUF

2C98 00 0B CALL @TRANSFILENAME ASCIIBUFF TO

2C9C 3F ASCIIBUFF

2CA0 16 04 MVI D,1 HLOGICAL FILE 1

2CA3 16 04 MVI D,4 NTYPE 4 FILE

2CAB 3A D7 29 LDA @RIVENO IGET DRIVE

2CBB CA 0C 01 MOV C,A

2CCD 21 4B 26 LHI H+FILEBUFFER JUSE DISK BUFFER 0

2CD9 0C 16 CALL @CREATE NEW DISK FILE

2CDE DA 8F 1C JC @DISKE-bit

2CFF 09 RET

2C97 E5 MAIN TRANSFER LOOP

2C9E 0A 01 FROM TAPE TO DISK

2C9F E5 INSERT LINE LENGTHS

2CA0 7E 30 WRITELoop MVI A+30H ISZER IN 'A'

2CA3 06 04 MVI B+BUF READER BYTES

2CAB 11 00 06 LHI D+buf MINIMUM LINE LENGTH

2CBE 21 A1 01 LHI D+BUFNT+ HSTART OF TEXT

2CC1 DA 07 1B CALL @FILLFA DEFAULT LINE 0000

2CCE 2B DCR H IPTR LSB LINE

2CD4 C5 FD 2C SKIP CALL TAPE 1 BYTE

2CD8 7B 2C CALL HNUMBCRN SKIP

2CD9 DA 8A 2C CNC SETFLG SET NZ TO

2CDB 2A CALL @FLAG HENG.

2CDE D2 A5 2C JNC SKIP IF NOT #

2CFF 17 15 HNUMBCRN HIST BYTE ALWAYS #

2C92 0B 03 MVI C+3 INEPEAT 3 TIMES

2C95 CD F0 2C WRITELPD CALL TAPE

2C99 D9 2C CALL HNUMBCRN

2CA0 D2 C9 2C JNC NUNBONE INOT NUMBER

2CBB CD 18 2D CALL SHFTLFT

2CC4 2F 2C MOV M,A

2CC5 0D 2C BCR C ICOUNT

2CC6 C2 84 2C JNZ WRITELPD IAND REPEAT

2CC8 CD F0 2C CALL TAPE

2CCB 23 NUMBDONE INX H IPOINT TO LOCATION

2CCD 88 20 CPI 20H IS IT SPACE?

2CCF 06 02 CNZ INSRTSC PCI 02H IIFS IT <CR>?

2CD0 CA 0B 2D CALL @TAPEBUF IPUT SPACE IN

2CDB 33 E4 2C INSRTSC HNUMBCRN 20H

2CDE 0F INX H IPOINT NXT LOC

2CDB 0C INR E INCREMENT LINELENGTH

2CDA 32 61 2E SETFLG STA LENFLAG ISL6 A NUMBER

2CDD E5 IFLAG NON-ZERO

2CE0 E9 RET

2CE2 E5 MOVE REST OF ONE LINE OF

2CE4 CD F0 2C CALL TAPE 1 BYTE

2CE7 0A 0B 2D JC @BUFISKORET IF <CR>

2CE4 77 CALL @TAPEBUF1 MOV M,A BYTE IN BUFFER
CONTINUED PAGE 8,
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 43°F to 92°F 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 realistic, but since we have the power problems and have to now and again do rewrites I wonder if a disc just for data to 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 be 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 111
Kathmandu, Nepal

Editor: I live in an area that the ambient room temperature runs between 50°F and 115°F and have had no trouble with my North Star single drive system. My North 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 disabuseful of those whose crime is 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.
3800 Bristol Road, Suite 167
Cornells 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 computer 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 what 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 "dismay." 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 might 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 with Stan that the 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 by a layperson, you should be able to handle it. As far as the other kinds of questions: deep in scope or not, let's 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-28 with a SELECTRICK II hooked up to my parallel port. I would like to hook a faster printer to the Sol, and have the two printers bode 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
210 Sprague Ave
So. Plainfield, N.J. 07080

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 1. The output is a Centronics 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 04538

CF/M TROUBLE WITH MICROPOLIS DRIVE

I am having trouble running CFM 11.41 and 11.31 thru my Sol. I have a 48K system and a Micropolis M5 drive. If anyone close to my area can give me some help, I'd appreciate it.

Paul Kittel
P.O. Box 1285
Loma Linda, Ca 92354
714-796-1388

WANTS TO PRINT AND SEE VIDEO AT SAME TIME

PORTUS,

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

If I want to print out the program or run and print it, I must go into (open up) the "Sol" and "Queue" to change the baud rates to make them compatible, and even if I did 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 printer and computer) each time I go from "W" to BASIC.

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

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

Jim Michaelis, Marketing
Dataram Publishing Co, Inc
P.O. Box 448
New Canaan, Ct. 06840
914-533-2263

Have you written Software?

SUPPORT YOUR CASSETTE AND DISK LIBRARIES
CF/M—NORTH STAR DIABLO DRIVER

Dear Mr. Solokow:

Please find $220.00 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 CFM/Sol Driver that would backup 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 55 plus 64 through 101. The rent is evidently standard, but I include it for completeness. The elimination of the deleted characters from the screen makes working with CFM immensely easier.

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

For those SOL owners who have a SOL-type 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: OCPAD EQU 195, 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, Mr. Starweather mentioned that he had adapted PLOT to the Northstar disk system and that he would make it available to the group. I am wondering whether anything has come of this.

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,

Kudos. B. Bucklin

[Alternate author sign: I believe the problem with the line type interface is that it doesn't allow enough time for the ribbon to lift after line feed, or it doesn't realize that the ribbon has dropped. If you physically hold the ribbon up, the problem can be avoided. - Stan]

1: "USER I/O ROUTINE AREA
2: ;
3: ;
4: ;
5: "EQUATES
6: MSIZE EQU 80 ;CHANGE THIS FOR DIFFERENT SIZE SYSTEMS
7: BIAS EQU (MSIZE-16)*1024 ;CHANGES BIAS AND ORG FOR SYSTEM
8: SSTAT EQU OFSH
9: SDA1 EQU 0FH
10: IOBIT EQU 3
11: USER ORG EQU 4400H+BIAS ;NOTE: 4500H
12: SOUT EQU OOC19H
13: DOUT EQU OOC1CH
14: SINK EQU OOC1FH
15: AINP EQU OOC22H
16: LF EQU 10
17: CB EQU 13
18: ;JUMP TABLE — CF/M JUMPS HERE FOR I/O
19: ;JUMPS MUST REMAIN HERE, IN SAME ORDER
20: INIT JMP INITR ;INITIALIZATION
21: CONSTAT JMP CSTAT ;CONSOLE STATUS CHECK
22: CONIN JMP CIN ;CONSOLE INPUT
23: CONOUT JMP COUT ;CONSOLE OUTPUT
24: LIST JMP LOUT ;LIST OUTPUT
25: PUNCH JMP PUNCH ;PUNCH OUTPUT
26: READ JMP READIN ;READER INPUT
27: ;
28: BEGIN USER DRIVER ROUTINES HERE
29: INITR XRA A STA IORTY ;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" 0
33: RET
34: CSTAT LOA CHAR ;SEE IF STATUS ALREADY GOTTEN
35: ORA A
36: JNZ STAT2 ;I F GOTTEN SAY SO
37: CALL SINP ;SET STATUS AND/OR CHARACTER
38: STA CHAR ;MOLD CHARACTER IF NOT
39: STAT2 WYI A ;CHARACTER WAITING
40: RZ
41: CMA
42: RET ;RETURN OFNH IF CHARACTER READY
43: 
44: CIN CALL CSTAT ;CONSOLE INPUT ROUTINE
45: ANA A
46: JZ CIN ;WAIT FOR CHARACTER
47: LOA CHAR
48: PUSF PSW
49: SUB A
50: STA CHAR
51: POP PSW
52: ANI CHAR ;STRIPE PARITY
53: STA LASC ;SAVE AGAIN TO CHECK FOR DELETE
54: RET
55: LASTAC DB 0
56: CHAR DB 0
57: FLAG DB 0
58: 
59: ;CONSOLE ROUTINE IGNORES MORE THAN ONE
60: CARRIAGE RETURN ON A LINE
61: 
62: COUT MOV A,C
63: CPI CR
64: JNZ LFCHK
65: PUSF H
66: LII 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: PUSF H
75: LII H,FLAG
76: MOV A,M
77: CPI M,G
78: JZ LFCHK
79: CPI 2
80: CPI 1
81: CALL OUTC
82: CPI C,CR
83: ENDCHK POP H
84: OUTC PUSF B
85: MOV A,C

CONTINUED ON NEXT PAGE.
HELP NEEDED: CENTRONIX/DIABLO/WORDWIZARD

DEAR TONY:

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

WE HAVE TWO SOL-SYS 111'S AND A CENTRONIX-700 PRINTER WE RUN BOTH PT DOS AND CP/M. (LIFEBOAT VS.) THE CP/M GIVES US SEVERAL CHOICES IN WORD PROCESSORS. THIS LETTER IS THE PRODUCT OF DIGITALS ED AND I.T.EX. WE HAVE SEEN WORD-MAN 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 USING A DIABLO OR EQUIVALENT THIS YEAR BUT BECAUSE 70 OR 80 PERCENT OF OUR COMMUNICATIONS IS IN-HOUSE AND DOES NOT REQUIRE LETTER-LIBRARY 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-WRITEL 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 NECK.

EARL J. DINAHM
941 N. RUSSELL
LA HABRA, CA 90631

ED: HELP, SOMEONE!

FORTH INTEREST GROUP

DEAR INTERESTED:

F.L.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.L.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:

<table>
<thead>
<tr>
<th>Service</th>
<th>US</th>
<th>OVERSEAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTH DIMENSIONS</td>
<td>$5.00</td>
<td>$9.00</td>
</tr>
<tr>
<td>PRIMER, 10 copies</td>
<td>$20.00</td>
<td>$26.00</td>
</tr>
<tr>
<td>USING FORTH, 200 pages, The best users manual</td>
<td>$20.00</td>
<td>$26.00</td>
</tr>
<tr>
<td>FIG-FORTH INSTALLATION MANUAL, 220-word glossary</td>
<td>$10.00</td>
<td>$13.00</td>
</tr>
</tbody>
</table>

All items sent post-paid. Enclose order with check or 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

HANGMAN / DUBIC FIX

Dear Tony,

In answer to the second item in Bob Stek’s letter to Santa (PROTEUS HEAD, Sept/Oct 79), there is a simple cure. Dubic and HansGen both expect to find the SOLOS START address in the HL resistor pair when they begin execution.

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

To make DUBIC flicker-free and executable as a PT DOS image file, perform the following:

1. Zip 0 to clear memory load DUBIC from tape.
2. Do a reset to SOLOS.
3. Enter 0720 121 00 00 C3 00 00
4. Execute RCBO to set back to PT DOS
5. IMAGE DUBIC, 14C0, 0, 725, 720

Do the same thing for HANGMAN except ENTER OEO 012. etc., and IMAGE HANGMAN, 14C0, 0, 725, 720

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

For those who do not have PT DOS, the syntax of the IMAGE command is:

IMAGE FILENAME FNAME:Blocksize:Ending address:End address:

Start or Executive address. The sequence above puts an LI X:O000H 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 LI instruction.

Also, I am interested in the AcePac Program; but would like to see them running before buying them. I’d like to visit with any user in Ohio, Kentucky, West Virginia or Indiana who would share his experiences with these programs.

Ben C. Starleton Jr.
2430 Summit Street
Portsmouth, Ohio 45662
10 REM *** 'PROGRAM PATCHDISK' ***
20 ' THIS PROGRAM WILL ACCESS AN EDITOR OR CPM SOURCE
30 ' FILE OF ANY LENGTH AND PATCH A RANDOMLY ACCESS
40 ' ED RECORD ON A CHARACTER BY CHARACTER BASIS. IT WAS
50 ' WRITTEN FOR 48K S0L-20 WITH D.D. NORTH STAR CP/M.
60 ' WITH MBASIC 5.0 OR LATER VERSIONS, VARIABLE
70 ' RECORD LENGTHS TO A MAXIMUM OF 128 MAY BE USED.
80 ' HENCE, A SEARCH MAY BE MADE FOR THE TARGET TEXT
90 ' USING THE MAXIMUM REC. LENGTH, THEN THE LENGTH
100 ' MAY BE DIVIDED AND THE RECORD NUMBER MULTIPLIED
110 ' LIKewise TO NARROW IN ON THE TARGET CHARACTERS.
120 ' THE TARGET RECORD IS VERIFIED ON RETRIEVAL.
130 ' FROM THE FILE AND ALSO AFTER PATCHES HAVE BEEN MADE.
140 ' THIS CODE MAY BE ADAPTED TO MBASIC 4.5 BY REMOVING RL
150 ' FROM LINE 560, AND MAKING IT A CONSTANT 128 IN
160 ' THE FIELD STATEMENT.
170 ' THIS PROGRAM IS PARTICULARLY USEFUL FOR MAKING SMALL
180 ' CHANGES TO EXTREMELY LONG ASSEMBLY LANGUAGE SOURCE
190 ' FILES WHICH OVERFLOW IN-MEMORY EDITORS, OR IN CASES
200 ' WHERE RECORD-BY-RECORD EDITORS ARE UNWIELDY. FOR
210 ' BREVITY NO EOF TEST IS MADE. LOF IS INVALID
220 ' BEYOND THE FIRST EXTNT.
230 ' Copyright 1980, Daniel S. Hunt
240 ' CLEAR
250 ' DEFPTR J, E, R
260 ' REM *** CONSTANTS
270 ' ESC = 27
280 ' WIDTH 64
290 ' REM *** KEY VARIABLES
300 ' AS : INPUT RECORD BUFFER STRING
310 ' BS : OUTPUT RECORD BUFFER STRING
320 ' XS : TARGET CHARACTER
330 ' X5 : TARGET CHARACTER
340 ' CS : MODIFIED CHARACTER
350 ' REM *** PROCEDURES
360 ' VERIFY INPUT RECORD
370 ' VERIFY OUTPUT RECORD
380 ' ASK USER IF O.K.
390 ' REM BEGIN...
400 ' PRINT CHR$(11)
410 ' PRINT *** RANDOM ACCESS TEXT PATCH UTILITY ***
420 ' PRINT ' Beware: There is no test for end of file!
430 ' INPUT ' ENTER FILENAME <OR STOP>': FS
440 ' IF FS = ''STOP'' THEN
450 ' PRINT *** STOP **** : END
460 ' INPUT 'RECORD LENGTH': RL
470 ' IF RL < 1 OR RL > 128 THEN PRINT 'REDO '; GOTO 490
480 ' ON ERROR GOTO 510 : ''TRAP FOR NON-FILE
490 ' OPEN ''': FI, RL AS FILE
500 ' PRINT *********** ERROR ****: GOTO 560
510 ' IF RL = 0 THEN PRINT *** ERROR ****: GOTO 560
520 ' IF RL < 1 THEN PRINT *** ERROR ****: GOTO 560
530 ' R$ = FS
540 ' PRINT R$ : RETURN
550 ' END
Dear Proteus,

Will PROTEUS help a person with a Vector Gratic Box with an 8080A CPU board and Vectorgrafic support boards. The I/O board is Tarbell as is the 32K ram boards. The terminal is a Smart.

The two floppy discs are eight inch Shugart 080/801 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 T. White
740 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 BOLGES/CUPS 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/NorthStar formats.

Daniel Hendricks
Irving, Ca

Editor: I had a long talk with one of the originators of the SAM76 language at the Second West Coast Computer Fair (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 Pi/M library, for example) develops the 16 bits of the answer by a series of 16 test and shifts with an optional 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 requiring 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 88 to 180 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 0-15 and columns in the range 0-63. Column address for tabbing on a printer result in numbers of the order 48–132. An analysis of random operands shows 32K cases where an answer of 2 results and only 2 where 32K results. Thus, the random operand case is biased towards smaller answers.

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

DIVIDE PUSH H ;SAVE DIVIDEND
CALL TOWSDE ;NEGATE DE
MOV L,H ;FORM HL/256
MV L,H ;FORM HL/256
CALL DIV ;DIVIDE INTO HL/256
MOV B,C ;PARTIAL ANS * 256
MOV A,L ;PARTIAL REMAINDER
MOV H,A ;PARTIAL ANSWER
POP H ;RESTORE ORIG. DIVIDEND
DIV B ;DIVIDE INTO HL/256
MV C,-1 ;INIT COUNT TO
DV INC C ;+1
DAD D ;SUBTRACT UNTIL
JC DV2 ;HL < O
CALL SUBDE ;RESTORE HL > O
RET

TOWSDE MOV A,H ;PERFORM 2'S COMP
CNA ;BY 7'S COMP
MOV D,A ;AND AND 1
MOV A,B
MOV A,E
MOV E,A
INX D
RET

SUBDE MOV A,L ;SUBTRACT DE FROM HL
SUB B
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 1/0 with a S10 Computer...written by Raj Mathia.

* SUB ROUTINE FOR EXECUTING DELAY LOOP
DELAY PUSH D C900 00
MVI D,0 C901 16 00

---

CONTINUED ON NEXT PAGE.
GREETINGS FROM MICROCOMPUTER RESOURCES!

I would like to comment on several things that I noticed in Volume 2, 45 of PROTEUS NEWS. Joe Maguire brought to light what he thought was a bug in PDOS. 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 entries. That is the way PDOS is supposed to work. The READ command will transfer all the bytes in a file. The image command files byte counts and store the 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,E800 will move it backwards 2000 hex bytes.

Get 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 than Lewis when it comes to repetitive work in PDOS. 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 tecns 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-NorthStar 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 EXTENDS EM-658 does indeed work with the HELIOS II very well.

The only board we found that is close is the PT NK-6A. 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 modern control package for the DC Hayes S-100 board that allows a SOL-NorthStar 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. The 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 $84.00 (plus postage.) The regular cost is $180, so this is a fair discount.

We will be working on software to connect the newest hard disk to PDOS (running with a Helios II on a SOL.) We know that there are lots of people looking for more storage, and it is an economic advantage to do so. We have been working on it as drive B, with a second reserved as C. The Helios II will think it is 2 and 3 (with no hardware changes...) The floppy would only be used for backup and short term storage. The hard disk plugs into the parallel port, so it does not fill your limited space SOL. Incidentally, we also intend to put PDOS on the Horizon, making software generate new 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 187-188
Tampa, FL 33612
(813) 977-5948
I HAVE THE SOURCE CODE!

At long last, I have actually gotten the source code disk for PTC's software, so 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: AL5-L, Business BASIC, GAMEPAK2, and a number of PTDOS commands.

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.

HELIDOS UPGRADE PROGRESS REPORT

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

TONY SEVERA REPORTS FOOL-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 him. If they haven't done so within the last 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 PTF documents listed in the "Proteus catalog" in prior issues (items having the prefix "OP" on their Proteus item number, such as "0637"). Make payment directly to Tony for the 0-issues. 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' address is Lewis Moseley, Jr., 2376 Glendale Ct NE, Conyers, GA 30015.

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" table and classified as placed by one John Glidea, 1246 Smallwood Dr, Columbus, OH 43220, purporting that "Sol chassises are being produced again with the computer to follow. I will guarantee fitness 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 agreement. He has no authority to manufacture rights. He says that this rumor has no basis.

Mr. Glidea 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. Its actual name is probably going to be "The Expander". It is designed by some English engineer who also made the Sol, Lee Felsenstein, so naturally there are similarities. But still, it's not the Sol.

ANYONE WANT ANOTHER SOL OR HELIDOS?

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 04KRA 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 Fair. 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 infential for a long time. Rest assured that all will receive 6 issues in 1980, as promised. Tony has delivered to me an unprocessed, issue, career-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 Balsec, the IDS 450 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 runs. Does anyone have 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 occasionally 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.
PATCHBC1.5 is a program to allow the reattributing of attribute protected files on PTDOS 1.3 (not 1.4). See text file HOWNF for more information on operation.

MESSAGE, IMESSAGE, 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 HOWNF for more information on operation.

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

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

FACTOR is a program to factor an integer into its prime. See text file HOWNF for more information.

KWIKSORT is a quicksort 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

SHELLSORT from Knuth

SORT is another sort suggested by Knuth which seems to approach the speed of the quicksort but may not have the disadvantages (when the file is in order, for example). Knuth rates the mathematical evaluation of this sort at 28 points, his maximum.

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

FIND was originally inspired by the WEE 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 = 9800h 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 corridors are set at 0 and the "cookie" is set at -1. Have the mouse increment any location he's at by 1 and then go to the lowest adjacent number. It works but looks funny sometimes. Also takes much less memory than FIND. Watch its behavior in the top right corner.

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

ACCOUNTS is a sample list of accounts used with BANKERS.

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

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...AMA6E...MMIND...BIORYM...LUNAR.1

EDITBH19 is a short program to allow the use of the Heath BH19 terminal with the PTDOS editor to use the terminals special function keys.

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

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

SPO,
SPB are simple serial-port output drivers. Their source codes are SPO.S and SPB.2, which may be reassembled, etc. through MAXDRVR macro.

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>SIZE</th>
<th>FIND+</th>
<th>FIND-</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>TYPE</td>
<td>SIZE</td>
<td>FIND+</td>
<td>FIND-</td>
</tr>
<tr>
<td>MESSM</td>
<td>16</td>
<td>15.5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>H-ADD</td>
<td>16</td>
<td>16</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ACCOUNT</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>ACCTG.2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ADD.WK</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>AMYMO</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>BANKERS</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>BIORYM</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>BOOTLDR</td>
<td>100</td>
<td>100</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CHASE</td>
<td>15</td>
<td>15</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CONTENT</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>COPY</td>
<td>15</td>
<td>15</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CREATFIL</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DIRECT</td>
<td>15</td>
<td>15</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EDITBH19</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>ELIZA</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EXPRSS</td>
<td>12</td>
<td>12</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>FACTOR</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>FILES</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>FIND</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

NAME | TYPE | SIZE | FIND+ | FIND- |

ROBOTS:C | T | 4
SER.RNDM | T | 4
SHELLSORT | 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 HLL SOME TIME AGO. THE PRESENT PROCEDURES NUMSTR AND ADD.WK, WORKING TOGETHER, PERFORM ONE OF OUR PRACTICAL APPLICATIONS WE USE THAT INCORPORATES THOSE AND OTHER ROUTINES IN OUR DAILY BUSINESS.

NUMSTR IS ADMITTEDLY LONG, BUT IT DOES A DIFFICULT JOB FOR US AND BECAUSE IT IS LOOSELY STRUCTURED IS EASILY MODIFIED AND HELP US TO GENERATE SOME OF THE MANY REPORTS WEgeführt UNDER THE SUPERVISION OF THIRTY-ONE RETAIL STORES FOR LIFE UNIFORM CO. IT IS EASILY IMPLEMENTED BY RELATIVELY UNSPECIALIZED HELP BECAUSE OF THE LEAD-THE-EYES HANDHELD, STRUCTURED WAY IT IS WRITTEN.

MUCH WILL BE 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 MIGHT LIKE TO USE HIS COMPUTER FOR AROUND THE HOUSE. OUR DATA-FILES ARE QUITE LARGE, BUT THE SHELL-NOT SORT Routines HANDLE THEM WITHOUT UNDE WANTS. I AM INCLUDING ONE OF OUR DATA-FILES SO THAT THE USER CAN RUN THE VARIOUS MODULES AND SEE HOW EASILY AND QUICKLY GOOD LOOKING REPORTS ARE GENERATED BY NUMSTR. THE MENUS NEEDS SIMPLE EXPLANATION. SIMPLY TYPE IN THE TITLE OF THE REPORT AND THE DATA-FILE NAME (THE ONE I INCLUDED ANY ACTUAL WORKING FILE) I HAVE ONLY CHANGED THE STORE NAME 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 FDOS.

PROGRAMS WORK TOGETHER. AFTER A NUMBER OF THE DATA-FILE IS CREATED IN NUMSTR, SUBSEQUENT WEEKS DATA IS ADDED TO THE FILES IN FDOS. IT RUNS VERY FAST AND THE WEEKLY DRAFT STORES ARE ADDED TO THE FILES IN MINUTES. THE HARD-COPY AND VDP REPORTS ARE ALL IN NUMSTR. I HAD TO INCLUDE THE DATA-FILE ON MY MINI COMPUTER AS THE PROGRAMS, SO IT WILL NOT BE CONSIDERED TO EITHER 'GET' IT TO YOUR UNI/L OR CHANGE THE PROGRAMS AT THE DATA-FILE SELECTOR MODULE AND DELETE LINE 1184 WHICH ADDS THE "A", "T" THING REALLY WORKS BEST WITH FDOS. DATA AND DATA-DISK RUNNING TOGETHER, WE BELIEVE THAT IT SAVES UNDERTAKEN AND الترك on UNI-ZERO'S HEAD TO USE THE SAME DATA BASE AS THE FIRST MENU, RESPONSE TO PROMPT WITH FOUR OR FIVE DIGIT INTEGERS WHEN IT ASKS FOR WEEKLY SALES. OUR STORE NUMBERS ARE A FIXED THREE NUMBER DIGITS AND THE STORE NAME TO TWENTY CHARACTERS IN LENGTH. THESE COULD OBTAIN BE ACCOUNT NAME AND NUMBERS IF YOU WISH TO ACTUALLY USE THE THING IN SOME PROFESSIONAL WAY.

UNLESS YOU HAVE A 780 SERIES CENTRONIX PRINTER, YOU WILL HAVE TO MAKE UP THE CONTROL CHARACTERS, NOT APPLICABLE TO YOUR PRINTER. CONTROL-L IS PAGE ADVANCE, CONTROL-K IS VERTICAL TAB, THE VERTICAL BAR GENERATES DOUBLE WIDTH LINES ON OTHER PRINTER MODES. THESE SYMBOLS WHERE USED AND IF PRINTER CONTROL WILL BE FOUND AT THE FOLLOWING LOCATIONS:


VERTICAL BAR WILL BE FOUND AT THE END OF LINES:

4170 - 4680 - 5166 - 5608 - 6984 - 7822 - 18528 - 11150

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 NEED A NUMBER OF INCENTIVE PLANS TO GENERATE ENTHUSIASM AND COMPETITION AMONG STORES, BEING ABLE TO SEE WHERE THE NUMBER RELATIVE TO OTHERS IS A BIG HELP. MORE IMPORTANTLY WE CAN ALSO GIVE A GLANCE IF A STORE IS HAVING PROBLEMS RISING UP OR DOWN THE LIST. WE HAVE LEARNED TO SPOT A PROBLEM BEFORE IT GETS TOO HARD BY PAYING ATTENTION TO THE REPORTS, THEN 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 PERSONNEL EXPENSES.

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

THE OPTIONAL VDP-ONLY DISPLAY IS VERY USEFUL TO US FOR FAST LOOKING AT THE SAME DATA AS BEFORE WE START THE HARD-COPY ROUTINE, IT IS SIMPLE TO ENTER DATA NUMBERS AT RANDOM AND WATCH THE WHOLE REPORT JUMP UP ON THE SCREEN. THAT BRINGS UP A VERY IMPORTANT POINT. WE MAINTAIN THE SAME STORE 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 STORE RECORD NUMBER AT WILL. IT REALLY MAKES AN EASY TASK OF RECALLING THE RECORDS OF ANY STORE WITH THE NEED FOR COMPLICATED SEARCH ROUTINES. TRY IT.

TO MAKE THE REPORTS SENSIBLE IT IS BEST TO INPUT ZEROS FOR FUTURE WEEKS WHEN INPUTTING 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 A LOOK AT HOW WE ARE DOING AS A REGION AND HOW OUR "BUDGETS" JUST LIKE THE INDIVIDUAL STORES.

THE IDEA BEHIND OUR ROUTINE IS THIS: AFTER CREATING A NEW DATA-FILE AND INPUTTING THE FIRST WEEKS DATA, SUBSEQUENT WEEKS ARE ADDED TO THE FILE BY RUNNIGN ADD.WK. IT IS VERY FAST, ONLY MINUTE IS REQUIRED TO UPDATE AN ENTIRE 38 STORES EACH ALMOST AUTOMATICALY AFTER MAKING THE PROPER SELECTION, SIMPLY PRINT 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 SET BY THE OPERATOR. THE POSITIONED PAGE IS JUST A PRINTING COVER TO THE PRINTER. WE COULD HAVE AVOIDED THE USE OF "CLOSED" AND "OPENING" PRINTER-DRIVER ROUTINES IF THE PROBLEMS WERE ENCOUNTERED AND MORE TIME WASTED THAN THE SINGLE SHEET OF PAPER WOULD BE.

THE STORE NUMBERS 68 THRU 258 ARE REAL TO US AND COULD BE CHANGED TO ANY CODE DESIRED. THE STORE NUMBERS ALSO ARE NAMES WHICH COULD BE ACCOUNT NAMES. THEY USUALLY REFER TO CITY NAMES FOR US AND I HAVE CHANGED THEM TO KEEP THEM OUT OF "TROUBLE WITH OUR BOOKS." THE SECOND ENTRY "B" IS THE BUDGET WHICH WILL BE USED 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.

IT IS A 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 WHIS-PROGRAMMERS WILL FIND MUCH MORE TO FAULT AND I'M SURE MANY SUGGESTIONS TO IMPROVE IT WILL BE OFFERED. I HOPE SO.

WE DO NOT LIMIT OURSELF TO CRITICISM, IN FACT WE ENCOURAGE IT. ANYONE FINDS MERIT IN IT WE WILL BE GLAD TO SHARE THE PROGRAMS WE USE IN OUR DAILY JOBS. ONE WAY OR THE OTHER WE WELCOME YOUR RESPONSES. R.N.

E.A. DUNNAB 941 N. RUSSEL LA HABRA, CA 90631
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
Goleans, Inc.
2608 8th Street
Berkeley, CA 94710

NEED A DELAY LINE OR nKRA?

John Edwards, 408 13th Street, #545, Oakland, CA 94612, 415/662-9394, 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 boards. He 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, 713/995-3453. 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.

Copyright (C) 1980 by Proteus. All rights reserved. Permission is hereby granted to reproduce any computer programs contained herein, provided that Proteus and the program's author are given credit.

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

James D. McElroy
2626 Crest Ave. North
Allentown, PA
18104
We start this issue with a large article written by our friend Fr. McGee entitled, "Understanding your SOL I/O Routines." I wish we had more of this type to be able to offer our readers! Everything Fr. McGee 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, Cali, 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 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, Cali, and recently changed their name to MicroSun Computers of Walnut Creek. Their demise was a long time coming. They were one of the first 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 capital. Being burned by P.T. and experiencing similar 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 28 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 487 PAPER TIGER, HEWLETT HIJ, HIJ, DIABLO, and printers that use either the CTS or DTR lines for handshaking. Look them over carefully. There are some neat things there, like FORMFEEDS for even klutsy old TTY's, and AUTOVIVING for you folks that like margins at the bottom of a page. If your printer is not represented, 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. I hope I can add something to your 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/NORTHERSTAR 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 SOLOS programs will continue to be available. The price is $10 for the MODIFIER II program and $5 for SOLOS, SOLOS 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/CUTTER tape or 8" soft sector CP/M disk.

Continued on next page.
UNDERSTANDING AND USING YOUR SOL: OUTPUT ROUTINES

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 SOLOS programs with only slight modification. I can supply these
on tape in Kansas City Standard 380 or 1200 Baud, complete with a
loader program for reading in the tape. The user only needs to know
the address of his tape read-byte routine.

Sorry to take up so much valuable space in Proctus, folks. I can
only hope that many of you find this stuff useful in one way or
another.

Sincerely yours,

Fr. Thomas McGaha
Fr. Thomas McGaha S.S.B.

PS: If you could print this stuff together as a block, it would
help. Thanks, Tony.

*** GENERAL OVERVIEW OF OUTPUT ROUTINES ***

SOUTH: C819: Uses pseudoport specified by OPORT (C887).
AOUTH: C81C: Uses pseudoport specified by “A” register.
VIDOUT: C814: SOL VIDEO DISPLAY (Pseudoport 0).
SIDOUT: C84A: SOL SERIAL PORT. (Pseudoport 1).
PRFOUT: C280: SOL PARALLEL PORT. (Pseudoport 2).
EPORT: C2D2: USER-DEFINED OUTPUT “DEVICE”.

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:

MOV A,B ;TRANSFER CHARACTER TO “B”.
CALL OUTPUT ;OUTPUT CHARACTER FROM “B”.
*** RETURNS WITH CHARACTER IN “A” ***

Some programs, such as CP/M, pass the character to be printed in the
“C” video device, 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
MOV B,C ;SAVE B,C REGISTER PAIR.
MOV B,C
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 SOUTH ROUTINE

SOUTH: C819: Pseudoport number recovered from OPORT (C887): 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.

SOUTH is the SYSTEMS OUTPUT routine. This is the primary entry point
for normal systems output. It is accessed by a CALL C819, and sends
out the character in the “B” register. The routine preserves all
registers except “A”.

SOUTH accesses one of four possible output devices, which will be
determined in detail later. For now, it is enough to say that the
VIDOUT routine selects the Serial Port, 1-the Parallel 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
C887. The binary value from 0-3 found at this location will be
loaded into the “A” register, and then control is passed to the AOUTH
routine (which is discussed next). The default value at OPORT is 0,
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 OX (where X is
a value from 0-3), will cause X to be the current pseudopost. It
should be noted that a RESET will cause the default value of 0 to be
loaded again.

The second way to load a value into OPORT is under program control.
For instance, the series of instructions MV A,3 / STA C887 would
cause the User-Defined Output Routine to become current. By the same
token, a program can find out what the current pseudopost is by
using the instruction LDA C887.

THE AOUTH ROUTINE

AOUTH: C81C: Enter with pseudostat # in “A”: Enter with character to be
output in “B”: Only the “A” register affected.

AOUTH stands for “A-OUTPUT”. The AOUTH routine is accessed by placing
a pseudostat code in the “A” register, the character to be output
in the “B” register, and calling C81C. The pseudostat code has the
following values:

0 = VIDOUT: C814, the SOL VIDEO DISPLAY
1 = SIDOUT: C84A, the SOL SERIAL PORT
2 = PRFOUT: C280, the SOL PARALLEL PORT
3 = EPORT: C2D2, the SOL USER-DEFINED ROUTINE

When the AOUTH routine is called, the value in the “A” register is used
to vector to one of the above mentioned pseudostats. It is
up to the user’s program to insure that the “A” register contains
a pseudostat number from 0-3. (The AOUTH program will discard all but
the two least significant bits, as it contains its own JNI 3
instruction).

*** PSEUDOSTATS ***

BESIDES USING SOUTH AND AOUTH, PSEUDOSTATS MAY BE DIRECTLY CALLED

; VIDEO DISPLAY: VIDOUT: C814

Enter with character to be output in “B”. Only the “A” register is
affected.

The Video Display is pseudostat 0, the default pseudostat. Besides
being available via calls to SOUTH and AOUTH, it may be directly
accessed via a CALL VIDOUT (CALL C814). It is suggested that either
SOUTH or AOUTH (with “A” set to 0), 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 88H ;MOVE CURSOR LEFT
; (WRAP MODE).
CLEAR 88H CTRL/A 88H ;CLEAR/SCROLL SCREEN.
RETURN 80H CTRL/A 88H ;CLEAR TO END OF LINE
; Go to START OF LINE.
HOME CURSOR 86H CTRL/A 86H ;HOME CURSOR
RIGHT-ARROW 91H CTRL/L 13H ;MOVE CURSOR RIGHT
; (WRAP MODE).
UP-ARROW 97H CTRL/W 17H ;MOVE CURSOR UP.
; (WRAP MODE).
DOWN-ARROW 9AH CTRL/2 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 00-3F HEX.
ESC CTRL/B XX ;Place the cursor at line XX. 00 is top line.
; XX must be in the range 00-3F HEX.
ESC CTRL/C ;RETURN character position in "B" (00-7F).
; RETURN line position in "C" (00-7F).
ESC CTRL/D ;RETURN memory address of cursor in "C" 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. 00=fastest, FF=slowest.
ESC CTRL/I XX ;SAME AS "ESC CTRL/A XX".

VIDEO DISPLAY PORT ASSIGNMENTS AND HOUSEKEEPING:

Video Display Status Port=ESTAT=FF
(An XPA A / OUT ESTAT will reset the screen scroll parameters).
Video Display Memory=CCB0-CFF0 organized as 15 lines by 64
characters.

Current Character Position: NCHAR=CB08
Current Line Position: LINE=C09
Beginning Of Text Displacement: BOUT=C0A
Display Speed: SPEED=CB08

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 &C30CH ;TEST FOR ESCAPE SEQUENCE.
ORA A
JNZ &C19H ;IF ESCAPFLAG IS ON, PROCESS SPECIAL.

TBLCHR MOV A,B ;NEED IN A TO STRIP PARITY BIT.
STA CHAR ;***SAVE ORIGINAL FOR LATER.
ANI 7H ;STRIP PARITY BIT.
MOV B,A
JZ &C83B
JMP &C827
;IF NULL, DO AN EXIT.
LXI H,&C278
;POINT TO SPECIAL CHARACTER TABLE.
CALL SRCH
;SEARCH TABLE & PROCESS IF FOUND.
JMP &C86B ;***FINISH INSIDE SOLGS.
NXT INX H ;SKIP TO NEXT CHAR, ENTRY.
INX H ; IGNORING ADDRESSES.
SRCH MOV A,M ;GET CHAR. FROM TABLE.
ORA A
JZ NOPE
JMP &C136
;IF LAST ENTRY, NOT SPECIAL.
CMP B ;TEST FOR A MATCH.
JNZ NXT ;GET READY FOR NEXT TRY.
JMP &C272 ;TRY AGAIN IF NO MATCH.
PUSH H ;SAVE ADDRESS IF MATCH FOUND.
CALL &C136
;REMOVE CURSOR.
XTHL ;GET DISPATCH ADDRESS TO HL.
JMP &C278 ;DISPATCH IT.
NOPE MOV A,B ;CHECK FOR DELETE.
CPI 7H
RZ ;DO NOT ECHO DELETE.
LDA CHAR ;***RECOVER ORIGINAL CHARACTER.
MOV B,A
JMP &C899 ;***FINISH WITHIN SOLGS.
CHAR NOP
;RAM STORAGE FOR CHARACTER*
;MAY BE ANYWHERE IN RAM*

*** NORMAL OR REVERSE VIDEO CHARACTER NOW PUT ON SCREEN ***

1; SERIAL PORT: SDOT: C04A

Enter with character to be output in "B". Only the "A" register is
affected. ("A"="A" upon return).

The Serial Port is pseudoport 1. Besides being available via calls
to SDOT and AOUT, it may be directly accessed via a CALL SDOT (CALL
C04A). It is suggested that either SDOT 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 SDOT 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
people use their SOL as a computer rather than as a terminal. The
following pair 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 20 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 20 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.
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. SC SIGNAL GROUND
8. CD CARRIER DETECT (IN)
9. CLO CURRENT LOOP OUT
10. LR1 LOOP RECEIVER 1
11. LR2 LOOP RECEIVER 2
12. DFR DATA TERMINAL READY (OUT)
21. 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 (9 bits, normal)
*** it is often wise to strip off the MSE before sending ***

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

<table>
<thead>
<tr>
<th>BIT</th>
<th>NAME</th>
<th>FUNCTION</th>
<th>ACTIVE DIRECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SCD</td>
<td>Serial Carrier Detect (EIA)</td>
<td>=Carrier On</td>
</tr>
<tr>
<td>1</td>
<td>SDSR</td>
<td>Serial Data Set Ready (EIA)</td>
<td>=Link OK</td>
</tr>
<tr>
<td>2</td>
<td>SPE</td>
<td>Serial Parity Error (UART)</td>
<td>=Error</td>
</tr>
<tr>
<td>3</td>
<td>SPEF</td>
<td>Serial Framing Error (UART)</td>
<td>=Error</td>
</tr>
<tr>
<td>4</td>
<td>SOE</td>
<td>Serial Overrun Error (UART)</td>
<td>=Clear</td>
</tr>
<tr>
<td>5</td>
<td>SCTS</td>
<td>Serial Clear To Send (EIA)</td>
<td>=Clear</td>
</tr>
<tr>
<td>6</td>
<td>SDR</td>
<td>Serial Data Ready (UART)</td>
<td>=Ready</td>
</tr>
<tr>
<td>7</td>
<td>STBE</td>
<td>Serial Asyl Buffer Empty (UART)</td>
<td>=Empty</td>
</tr>
</tbody>
</table>

The SOL SERIAL ROUTINE follows:

PRN: IN STBE, 1; GET SERIAL STATUS WORD;
RNL: 1; TEST FOR SERIAL DATA READY.
JNC: SENDT: 1; CARRY SET=OK TO SEND.
MOV A,B; 1; WE SEND FROM "A".
OUT: 1; 1; SEND CHARACTER.
RET: 1; ***CHAR. IS IN "A" AND "B".

NOTE: Instead of ANI STBE, RAL was used since it saved 1 byte.

2: PARALLEL PORT: PROUT: C265

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 C265). 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=STAP=FA
Status Bit=FXDR=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. IF OUTPUT ENABLE (IN) =ENABLED
4. IDP *DATA READY (IN) =READY
5. *PAK *OUTPUT ACKNOWLEDGE (OUT) =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)
10. ID3 OUTPUT DATA, BIT 3 (IN)
11. ID2 OUTPUT DATA, BIT 2 (IN)
12. ID1 OUTPUT DATA, BIT 1 (IN)
13. ID0 OUTPUT DATA, BIT 0 (IN)
14. US UNIT SELECT (OUT)
15. OE OUTPUT ENABLE (IN) =ENABLED
16. *XOR *EXTERNAL DEVICE READY (IN) =READY
17. *OL *OUTPUT LOAD (OUT) =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 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 ROUTINE:

PRN: IN STAP; GET STATUS WORD,
ANI FXDR; TEST PARALLEL PORT ONLY.
JNZ: PROUT; WAIT UNTIL OK TO SEND.
MOV A,B; SEND FROM "A".
OUT: 1; SEND CHARACTER.
RET; ***CHAR. IS IN "A" AND "B" ***

3 USER-DEFINED PORT: ERROUT: C272

Enter with character to be output in "B". Only the "A" register is affected. Any OTHER characteristics are up to the USF, 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 ERROUT (CALL C272), 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 ERROUT 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 \texttt{U400F} (C882-C883). There are two ways of accomplishing this: The user may type \texttt{SE CO=XXXX}, where XXXX is the address of the user-defined custom routine, or he may load the address in user program control. For example, the instruction \texttt{LXI B,1234 / SHLD C802 would cause the address 1234 to be loaded at C802-C803. (Addresses are loaded in reverse order... Low order at C802, and High order at C803.)

Remember also that it is up to the user to make the current pseudoport-1 when using \texttt{SOUT}, and to make \texttt{"A"-3} when using \texttt{AOUT} to access the custom output routine. After a reset, the \texttt{OPORT} and \texttt{UGORT} are set to \texttt{0000} respectively. The \texttt{FPRINT} routine will assume an error has occurred whenever it finds \texttt{0000} at \texttt{U400F}. This means that \texttt{0000} 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 line feeds 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 8 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 \texttt{SOUT} and \texttt{AOUT}. This is not really a problem, since custom routines may be selected by \texttt{SOUT} and \texttt{AOUT}. However, custom routines may be used to allow one custom routine to load into another custom routine. I routinely use \texttt{CONTROL/X} as a special character that causes a given program 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 \texttt{SOUT} and \texttt{AOUT} is a powerful feature.

This material presented courtesy of:
Fr. Thomas McAlahee
202 Union Ave.
Paterson, NJ 07502
(201)-595-8888

*** SAMPLE DRIVERS FOR IDS 442 PAPER TIGER ***
*** ASSUMES YOU HAVE IT CONNECTED TO SERIAL PORT ***
*** WRITTEN BY FR. THOMAS MCCAHEE ***
*** DON BOSCO TECHNICAL HIGH SCHOOL ***
*** 202 UNION AVE., PATERNER, NJ 07502 ***
*** JUNE 9, 1980

******* EQUATE TABLE **********

\texttt{PAGE EQU 50 \quad \#FIFTY LINES PRINTED PER PAGE.}
\texttt{MAX EQU 65 \quad \#MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).}
\texttt{PRMFD EQU ACU \quad \#CODE FOR FORMFEED.}
\texttt{SRFST EQU 00H \quad \#SERIAL STATUS PORT.}
\texttt{SDROT EQU 004AH \quad \#SERIAL OUTPUT ROUTINE.}

\texttt{VDMOT EQU AC54H \quad \#VIDEO OUTPUT ROUTINE.}
\texttt{LF EQU BAH \quad \#CODE FOR LINE-FEED.}
\texttt{CR EQU 0DH \quad \#CODE FOR CARRIAGE-RETURN.}

*** FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER ***
*** INCLUDES HANDSHAKING, ENTER WITH CHARACTER IN "A" ***
*** EXIT WITH CHARACTER IN "A" AND "B" ***

\texttt{TIGER: IN SERST \quad \#CHECK IF PRINTER READY.}
\texttt{ANI 22H \quad \#CHECK BOTH CTS AND DSR.}
\texttt{JNZ TIGER \quad \#WAIT IF BUSY.}
\texttt{CALL SDROT \quad \#PRINT CHARACTER IN "A."}
\texttt{RET \quad \#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 50 LINE FEEDS A FORMFEED OCCURS. ***
*** THIS KEEPS LISTINGS NEAT. ***
*** ANOTHER ADDED FEATURE IS THAT THE VIDEO SCREEN ***
*** WILL ECHO WHAT IS BEING PRINTED. ***

\texttt{LIST: PUSH B \quad \#SAVE B.C.}
\texttt{MOV B,C \quad \#CP/M HAS IT IN "C" & SOL NEEDS IT IN "B".}
\texttt{VIDEO: CALL VDMOT \quad \#PRINT TO VIDEO FIRST.}
*** ELIMINATE "CALL VDMOT" IF YOU WANT PRINTER ONLY.
*** I LIKE TO SEE THE STUFF ON THE SCREEN AS WELL.
\texttt{PRINT: IN SERST \quad \#CHECK IF PRINTER READY.}
\texttt{ANI 22H \quad \#CHECK BOTH CTS AND DSR.}
\texttt{JNZ PRNTR \quad \#WAIT UNTIL CLEAR.}
\texttt{MOV A,0 \quad \#CLEAR PORT.}
\texttt{CPI FRMFD \quad \#CHECK FOR FORMFEED.}
\texttt{JZ PFEED \quad \#PRINT USING SERIAL PORT.}
\texttt{CPI LF \quad \#IS IT A LINE-FEED?}
\texttt{RNZ \quad \#IF NOT.}
\texttt{PAGER: LODA COUNT \quad \#GET CURRENT COUNT.}
\texttt{DCH A \quad \#REDUCE IT BY 1.}
\texttt{STA COUNT \quad \#STORE NEW COUNT.}
\texttt{JNZ NOPAGE \quad \#IF NOT END.}
\texttt{FFED: LODA LENGTH \quad \#GET NEXT PAGE LENGTH.}
\texttt{STA COUNT \quad \#IF NOT END.}
\texttt{MV B,FRMFD \quad \#PRINT FORMFEED.}
\texttt{CALL SDROT \quad \#DO A FORMFEED.}
\texttt{NOPAGE: POP B \quad \#RECOVER ORIGINAL R.C.}
\texttt{MOV A,C \quad \#CP/M LIKES IT IN "A" TOO.}
\texttt{RET \quad \#RETURN.}

\texttt{COUNT DB PAGE \quad \#LINES LEFT ON THIS PAGE.}
\texttt{LENGTH DB PAGE \quad \#LINES PRINTED PER PAGE}
*** NOTE THAT "PAGE" MUST BE < THAN OR = "MAX".
\texttt{AX}

*** SAMPLE DRIVERS FOR DIABLO PRINTERS ***
*** ASSUMES YOU HAVE IT CONNECTED TO SERIAL PORT. ***
*** WRITTEN BY FR. THOMAS MCCAHEE ***
*** DON BOSCO TECHNICAL HIGH SCHOOL ***
*** 202 UNION AVE., PATERNER, NJ 07502 ***

********** EQUATE TABLE **********

\texttt{PAGE EQU 50 \quad \#FIFTY LINES PRINTED PER PAGE.}
\texttt{MAX EQU 65 \quad \#MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).}

Continued on next page.
UNDERSTANDING AND USING YOUR SOLO OUTPUT ROUTINES

**FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER.***
**INCLUDES HANDSHAKING VIA FTA AND ACK CODES. ***
**ENTER WITH CHARACTER IN "B" ***
**EXIT WITH CHARACTER IN "A" & "B". ***

**DIABLO:***
LDA CHRCNT ;UPDATE CHARACTER COUNT.
STA CHRCNT ;STORE UPDATED CHARACTER COUNT.
JNZ OK ;IF OK, ACTIVATE PRINTER.
MV1 A,BUFF ;OTHERWISE RESET CHARACTER COUNT.
STA CHRCNT ;STORE CHARACTER COUNT.
MV1 B,BUFF ;PREPARE TO SEND ETX.
CALL SDROT ;SEND IT.

**WAIT:**
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.
MV1 B,BUFF ;RECOVER ORIGINAL CHARACTER.
POP H ;PRINT ORIGINAL CHARACTER.
OK ;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 AUTO-PAGING. ***

**LIST:**
PUSH B ;SAVE ORIGINAL.
MOV A,B ;CP/M HAS IT IN "B", SOLOS NEEDS IT IN "A".
CALL VROM ;***MOVE IF VIDEO ECHO NOT WANTED.
MOV A,B ;CP/M FORMFWD.
JZ FFED ;HANDLE FORMFWD SPECIAL.
LDA CHRCNT ;UPDATE CHARACTER COUNT.
DCA A ;STORE UPDATED CHARACTER COUNT.
JNZ OK ;IF BUFFER NOT EXCEEDED.
MV1 A,BUFF ;RESET CHARACTER COUNT.
STA CHRCNT ;TO BEGINING.
MV1 B,ETX ;SEND OUT ETX.

**WAIT:**
JZ WAIT ;WAIT UNTIL ACK RECEIVED.
ANI 7FH ;IGNORE ALL BUT ACK.
JNZ WAIT ;GET ORIGINAL BACK INTO "B".
CALL SDROT ;PRINT ORIGINAL CHARACTER.
CPI LF ;TIME TO END THIS PAGE?
JZ PRINT ;IF NOT, THEN PRINT IT.

**PAGER:***
PUSH B ;SAVE ORIGINALS.
MOV B,C ;CP/M HAS IT IN "C", SOLOS NEEDS IT IN "B".
CALL STAT ;CHECK TO SEE IF BUSY FIRST.
JZ PRINT ;IF NOT, THEN PRINT IT.
CPI BF ;IS IT REALLY BUSY?
JNZ PRINT ;IF FALSE ALARM, PRINT "B".
REDY ;CALL REDY.
CPI READY ;CHECK TO SEE IF IT IS READY.
JNZ REDY ;IGNORE ANYTHING ELSE.
PRINT ;CALL SDROT ;PRINT ORIGINAL CHARACTER.
DEC ;DONE.

**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 STAT ;CHECK TO SEE IF BUSY FIRST.
JZ PRINT ;IF NOT, THEN PRINT IT.
CPI BF ;IS IT REALLY BUSY?
JNZ PRINT ;IF FALSE ALARM, PRINT "B".
REDY ;CALL REDY.
CPI READY ;CHECK TO SEE IF IT IS READY.
JNZ REDY ;IGNORE ANYTHING ELSE.
PRINT ;CALL SDROT ;PRINT ORIGINAL CHARACTER.
DEC ;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 AUTO-PAGING. ***

**LIST:**
PUSH B ;SAVE ORIGINAL.
MOV B,C ;CP/M HAS IT IN "C", SOLOS NEEDS IT IN "B".
CALL STAT ;CHECK TO SEE IF BUSY FIRST.
JZ PRINT ;IF NOT, THEN PRINT IT.
CPI BF ;IS IT REALLY BUSY?
JNZ PRINT ;IF FALSE ALARM, PRINT "B".
REDY ;CALL REDY.
CPI READY ;CHECK TO SEE IF IT IS READY.
JNZ REDY ;IGNORE ANYTHING ELSE.
PRINT ;CALL SDROT ;PRINT ORIGINAL CHARACTER.
DEC ;DONE.

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 FORMED
JZ FFED
; FORMFEEDS ARE HANDLED SPECIAL.
CALL SDRO T
; PRINT ALL OTHERS.
CPH LF
POP B
; PREPARE FOR RETURN.
RZ
; RETURN IF NOT A LINE-FEED.
LDA COUNT
; UPDATE LINE COUNT.
DCA A
; STORE UPDATED COUNT.
MOV A,C
; CP/M LIKES IT IN "A" TOO.
RZ
; RETURN IF PAGE NOT DONE.
FFED:
LDA LENGTH
; RESET PAGE COUNT.
STA COUNT
; SEND UNCONDITIONAL FORMFEED.
CALL SDRO T
; RECOVER ORIGINALS.
POP B
; ALL DONE.
RET
COUNT DB PAGE
; LINES PER PAGE (PRINTED).
LENGTH DB PAGE
; MAXIMUM NUMBER ALLOWED TO PRINT.

MVI A,MAX
; RESET LINE COUNT AT PAGE BOUNDARY.
STA COUNT
MOV A,B
RET
FFED:
LDA COUNT
; RETURN WITH NEW PAGE SET.
POP B
; GET CURRENT PAGE LENGTH.
LPSW:
; SAVE # OF LINES TO DO.
MVI B,A,L
; SEND OUT A LINEFEED.
CALL SDRO T
; APPEND THE NECESSARY NULLS TOO.
POP PSW
; RECOVER # OF LINES TO GO.
DCA A
; UPDATE IT.
JNZ FFED
LPSW
POP B
MVI A,MAX
; FOR A FULL PAGE LENGTH.
STA COUNT
MOV A,B
RET
; COPY "B" INTO "A" FOR CONSISTENCY.
; AND NOW WE IS DONE.

COUNT DB MAX
; ALLOWS CALCULATED FORMFEEDS.

*** 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
*** 202 UNION AVE., PATERSON, N. J. 07503

************ EQUATE TABLE ************
PAGE EQU $58
; FIFTY LINES PRINTED PER PAGE.
MAX EQU $56
; MAXIMUM PAGE SIZE (INCLUDING BLANK LINES).
FFED EQU $01
; CODE FOR FORMFEED.
SDRO T EQU $0494
; SERIAL OUTPUT ROUTINE.
VDMOT EQU $0496
; VIDEO OUTPUT ROUTINE.
LF EQU $0104
; CODE FOR LINE-FEED.
NULLS EQU $10
; NUMBER OF NULLS AFTER LF.

*** FIRST EXAMPLE IS A SIMPLE SOLOS-TYPE DRIVER. ***
*** ENTER WITH CHARACTER IN "B", ***
*** EXIT WITH CHARACTER IN "A" & "B", ***
*** FORMFEED ACCOMPLISHED VIA CALCULATED LINE-FEEDS. ***

TTY:
MOV A,B
; CHECK FOR FORMFEED.
CPI FORMED
JZ FFED
; HANDLE FORMFEEDS SPECIAL.
CALL SDRO T
; PRINT CHARACTER IN "B".
CPH LF
; THEN CHECK TO SEE IF IT WAS A LINEFEED.
RZ
; IF NOT, WE IS ALL DONE.
CALL NULL
; IF SO, SEND OUT SOME NULLS!
PAGER:
LDA COUNT
; STORE UPDATED LINE COUNT.
DCA A
; UPDATE LINE COUNT.
MOV A,B
; PREPARE FOR RETURN.
RZ
; RETURN IF ALL DONE.

LIST:
LPSW
MVI B,A,L
; SAVE # OF LINES TO DO.
CALL VDMOT
MOV A,B
; HANDLE FORMFEED SPECIAL.
CALL SDRO T
; PRINT CHARACTER IN "B".
CPF LF
; THEN CHECK TO SEE IF IT WAS A LINEFEED.
POP B
; PREPARE FOR RETURN.
RZ
; IF NOT, WE IS ALL DONE.
CALL NULL
; IF SO, SEND OUT SOME NULLS!
PAGER:
LDA COUNT
; UPDATE LINE COUNT.
DCA A
; STORE UPDATED LINE COUNT.
MOV A,B
; PREPARE FOR RETURN.
RZ
; RETURN IF ALL DONE.

MVI A,MAX
; RESET LINE COUNT AT PAGE BOUNDARY.
STA COUNT
MOV A,B
RET

FRACT:
LDA COUNT
; RETURN WITH NEW PAGE SET.
POP B
; GET CURRENT PAGE LENGTH.
LPSW:
; SAVE # OF LINES TO DO.
MVI B,A,L
; SEND OUT A LINEFEED.
CALL SDRO T
; APPEND THE NECESSARY NULLS TOO.
POP PSW
; RECOVER # OF LINES TO GO.
DCA A
; UPDATE IT.
JNZ FFED
LPSW
POP B
MVI A,MAX
; FOR A FULL PAGE LENGTH.
STA COUNT
MOV A,B
RET
; COPY "B" INTO "A" FOR CONSISTENCY.
; AND NOW WE IS DONE.

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 AUTOPAGEING. ***

LIST:
LPSW
MVI B,C
; "A" CONTAINS THE NULL CHARACTER NOW.
MVI C, NULLS
; "C" CONTAINS THE NULL COUNT.
NLOOP:
CALL SDRO T
; SEND OUT A NULL.
DCA C
; UPDATE NULL COUNT.
JNZ NLOOP
; SEND OUT PROPER AMOUNT.
POP B
; RECOVER ORIGIANLs.
RET
; ON WITH THE SHOW.

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
*** NULL GENERATOR ***
NULL:  PUSH B          ;SAVE ORIGINALS.
       MVI A,B         ;"B" CONTAINS THE NULL CHARACTER NOW.
       MVI C, NULLS    ;"C" CONTAINS THE NULL COUNT.
NLOOP: CALL SDRTOT  ;SEND OUT A NULL.
       DCR C          ;UPDATE NULL COUNT.
       JNZ NLOOP      ;SEND OUT PROPER AMOUNT.
       POP B         ;RESTORE ORIGINALS.
       RET
COUNT DB PAGE    ;ALLOWS CALCULATED FORMFEEDS.
                   ;AND ALSO AUTOPACING.
```

ADVERTISEMENTS: I have custom USER routines for NORTHSTAR that include drivers for a wide variety of printers operating under control of a SOL-28. 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.
LOAD is converted to a COMMA for numeric keypad use.
USER MAY EASILY ADD HIS OWN CONVERSIONS.
CONTROL/P can be used to cause PRINTERS to echo VIDEO!
CONTROL/V can be used to return to VIDEO-ONLY mode.
CONTROL/VP are available at all times, even during listings!
OUTPUT DEVICE may be selected as device #1.
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.
PRINT can be switched on and off even during a listing.

PRINTERS SUPPORTED

```
SERIAL TELETYYPE WITH NULLS AFTER LINEFEED

NSTARSW14 Connects via SOL Serial Port #1. Provides delay following a linefeed by sending out a user-defined number of nulls. Comes set for 16 nulls. FORMFEED is accomplished with linefeeds. MAY BE USED WITH ANY SERIAL PRINTER.
```

```
SERIAL WITH HANDSHAKING VIA CLEAR-TO-SEND

NSTARSCH 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 20 of the RS232 connector. Use for PAPER TIGER, etc.
```

```
SLOWED SERIAL via SOL Serial Port #1

NSTARLRS 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 1620 (or equivalents such as IPM 1622)

NSTARH10 Connects via SOL Serial Port #1. Allows use at ANY baud rate without loss of data. Uses ETX (#3) and ACK (#6) 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 $10 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, $20 elsewhere.

As currently written, DOS is assumed to have an origin at 2000 with the USER AREA starting at 2800. 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 CONT are located (they start at 2800 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 $10. Add $5 if outside continental U.S.A., and $5 if it includes customizing of any kind.

Program is supplied on SORDS/CUTER 1200 baud cassette tape in BININARY FORMAT. THE SOURCE CODE is provided as HARD COPY.

Send check or money order to:
   Fr Thomas McGhee
   202 Union Ave.
Patterson, New Jersey 07582

For inquiries of any kind concerning this or any of my other software or services, please include a stamped, self-addressed envelope.
REPLACE MONITOR 2708 WITH 2716

Gentlemen:

The Sol 20 Monitor Rom (Solos-1) may be easily replaced with a 2716 Eeprom. Enclosed are the hardware modifications on the PC board necessary to do this. The Object Code located at CO00-C7FF is burned into the Eeprom prior to making the hardware changes, or can be done on a second piece of equipment.

Sincerely,

Howard E. Chace

---

Replace Monitor Rom with 2716 Eeprom

1. Remove Trace Connecting Pin 19 to 12V
2. Remove Trace Connecting Pin 21 to C0
3. Install Jumper Connecting Pin 19 to C0
4. Install Jumper Connecting Pin 21 to 5VDC

---

Diagram:

- [Diagram of circuit modifications]
  - Note: Version 1 (2708 Eeprom Version)
  - Pinout which is different from Rev 0 (2716 Masked Rom Version)

---

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2708</td>
<td>Monitor Module</td>
</tr>
</tbody>
</table>
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 PVT 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 transferred 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 PENVIL that operates at both 300 and 1200 baud. Operating at 300 baud seems 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 Rock 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 baud. The replacement part is SIGMA 191TE1C-55. 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,00H  
OUT M1  
get high speed bit

These instructions switch both the SOL and the modem to low speed:

MVI A,01H  
OUT M1

To switch everything back to high speed, use the following:

MVI A,00H  
OUT M1

get high speed bit
OUT SBD
MWI A,MB
OUT MST
switch computer
get high speed bit
switch modem

The equates used for the above are:

MST EQU 0F6H
SBD EQU 0F8H
SHI EQU 06H
SLO EQU 06H
MWI EQU 10H
MLO EQU 06H

Modem Status port
Speed relay port
Sol HI speed bit
Sol LO speed bit
Modem HI speed bit
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
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 PTDDOS 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 tracor. 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 PTDDOS with the file to be WIZed on
a disk in slot 1. Then remove the PTDDOS disk from slot 0 and
insert the WordWizard systems disk. Don't bootload WordWizard
yet. While still in PTDDOS type:

WIZ filename/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 PTDDOS 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
PTDDOS 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 Solos. 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 Solos.

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,

Michael A. McKelvey
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 8, La Grande, OR 97850. (541) 963-5432. Herbert E. Swett; Editor and Publisher.

LOOKING FOR I-CHING

I have a Sol-28 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 Edgewood 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 analyzer 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 Meyer, 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 have a fix for this "bug" it would be greatly appreciated, and would be communicated to others who have the same problem. Joseph Freal, 22840 S.W. 164th Avenue, Goulds, Florida 33173.

(Editors note: Please send it also to PROTEUS so we can publish it)

WANTS TO INTEGRATE 2-BB 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 2-BB into their machine and still run the majority of their software. I know of a "2-BB" conversion whereby a 2-BB CPU may be plugged into the backplane, but is there anything (like the Neutronics conversion for IMSAI boards) which works for the Sol?

(RMSAI used standard INTEL support chips but Processor Technology didn't.) I'd like to use my 2-BB macro assembler but can't.

I have looked at 2-BB 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 clickers, warts, or crashes) in my Sol/2M 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: What's the story?

Sincerely,

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 System. I have two Sol-20's and last fall equipped 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 SOLBIO'S) is provided which believe it
or not means that a Sol owner who wants a disk system now can actually buy a "turnkey" one. I can testify that you literally can unpack, plug in, turn on, type EX D980 (CR) and be in CP/M.

My memory boards consist of PTC 16KRA and 32KRA, a WAMEM M2M 16K oard populated with 2114 chips and an SD Salem Expandorand 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 trouble 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-280 system has been flawless. Since their installation in October, 1979 there has been no down time. The SOLBICS contains drivers for disk I/O and Solos CRT and keyboard drivers for other I/O, Skeleton drivers for all 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-280 systems with any of the Miroco Peripherals drives. Thus you can have 400K bytes to 1.6Mbytes on a dual drive system depending on whether you use the 851, 852, 859 or 859 MFI drives.

Charles H. Stembridge

SOL BASKET-CASE NEEDS ASSISTANCE

I just acquired a Sol-28 (basket-case, but that's the only kind I could afford) and a friend told me about PROTEUS (Capital) 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 apparently random characters. A keyboard reset always clears this and results int he SOLDO prompt.

2. Occasionally (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 TREKB 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 2), and the game never begins.

If I run the ESY Diagnostic RAM Test the memory board checks out OK and TREKB 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 putting 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 LESSE "For Sale" Ads. These are nice, but not every issue. (Maybe once or twice a year.)

I have only had my Sol-28 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 $8000(H) and it will fill the remaining memory with whatever and then dump to the screen to check for errors.

$8000 21
$8001 10 00 16 hi bytes of start address
$8002 38
$8003 00 byte value (use $0,55,AA,6FF)
$8004 01
$8005 77 23 7C FF
$8006 90 hi byte of last mem location plus 1
$8007 C2 03
$8008 C0 hi byte of where loaded
$8009 84 04
$800A 84 04 run back to Solus
1. Enter at $8000(H)
2. EX 0
3. DU 0-FFFF (36K in my case) To change byte value EN: 04 CR

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 newsletter is a bargain.

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. McIeghe is a constant contributor and the problem lies with the fact that except for him and several others; they are the only major contributors. 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 and they also 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 for us 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, the board had already been sold. The second example was when such items were available, 90% of the boards were not available anymore. I had to sell them on a first call, first served. Out of this sale of 40, 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 was a "professional" magazine, you wouldn't get the personalized service. We may be slow but we care!

RECEIVING LEE PELSONSTEIN

I just received my April/May issue of Proteus/News and fund the information on page 19 about the work that Lee Pelsonstein has mapped out very interesting, particularly about relocating Solos to upper memory...Keep us informed! Also, has anyone adapted the 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 to consider for national and international publication, please contact:

Stephen Rosch
Games Editor
Hayden Publishing Co., Inc.
56 Essex St.
Rockelle 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. Rosch at the above address.

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 the point.

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 fair, concerning Ham Radio Applications. At that time, you asked if I might look over the literature, perhaps borrow some 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 haven't had a chance to write back.

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 to do this would have been an 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 96K 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 $50 and $500. Have you ever heard of such a rip-off??? Apparently the supply company is in New Jersey somewhere. Every other member 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, I'm back in business, and if I have any success with 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.

While I can see 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 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 regards to the matter Stan I am also interested in Source Code for AB_-2 revision B, and TREK-50 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 Assembler 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 this 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 its 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 V. Kandzia

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 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 dont 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 with outley.
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. Warten REG
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 (213) 711-9000 (area code 415)
home phone (415) 754-8943

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 ever have disks, this hobby is something to have to pay for itself.

Second, many thanks to Bob Werner for sharing his fix for those %SOS!! 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 EBCBASIC 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 IBM'S Coursewriter BASIC but just two in XREF and it prints a list of the variables in alphabetical sequence and the line numbers each is used in. Just like the one produced by AS-8 when you assemble a program. I would be very grateful to anyone who has done this for PTC EBCBASIC. 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

Robert W. Heerdink
Matrix Operations

One of the most powerful features of PT EOMBASIC is the group of MAT statements. Many non-mathematical persons are not aware of how useful they can be in speeding up business-like 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 3030 arrays the running time is 14.6 seconds. Substituting MAT Z=X+Y for the loops cuts the time to 1.8 seconds. Try to watch that with an Apple or TR-60!

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 EOMBASIC manual 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 once again. This newsletter was partially done by our former editor, Tony Reeves. 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 ACS-6 source code, finally. More in the next issue.

Len
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 "DATAMARK". This is a 5-character tape label in the following format: AYMDY, 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, 3:March, 4:April, 5:May, 6:June, 7:July, 8:August, 9:September, 10:October, 11:November, 12:December), and DD is the day of the month, plus:

SAVE 05322 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 alone with a description of what is 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 addressess 0 FFFF to write an empty file is not as well known as it should be. If you are storing files from PT-EDIT, the command 00032300FE will have the same effect.

Syntax Corporation
4500 W. 72nd Terrace
Prairie Village, Kansas 66208
Phone (913) 362-8869

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
Editor & Publisher
The Financial Systems Report

Encl.

[If you are interested in financial planning, write to these people for a sample newsletter. - Ed]
CP/M USERS GAIN IMPROVED KEYED FILE SUPPORT

Van Nuys, Calif. — Micro Applications Group, a microcomputer systems house here, has introduced MAGSAM IVm, a new high-performance version of their keyed file management system. This enhanced system combines the features and capabilities 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 initiated 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, MAGSAMX 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 Microcosm 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 Caldesa Avenue, CA 91406.

********************************************************************

MAIL ORDER & MEDIA ANALYSIS SOFTWARE FOR SOL

MAGMANK ADVERTISING RATE CARD ANALYSIS: DISK VERSION
Requires 48K Sol, North Star format CP/M, two 8" 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. Analyzes: cost per thousand, unit frequency and continuity discounts, fractional page impost 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: $800.

(A sample disk of 7 representative rate analyses, without filling 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 response to produce "range" forecasts and pinpoint break-even sales levels. CBASIC2 run-time package, user documentation included.

PRICE: $100.

DANIELS HUNT / PUBLISHING STRATEGIES / 16 Escondido Court / Newport Beach, CA 92663

EDITOR: FOR THOSE WHO MISSED IT!

TAD Enterprises
P.O. Box 257
Hamlin, IL 60429
Well, you're 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 you've kept 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 try and help you to provide you with what you need. It can be as little as information on types, costs and capabilities as to just assistance in setting up and operating a system of your own.

When business people think about computers several questions may come to their minds:

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 do 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 200 periodicals, trade magazines, user journals and books being released every month 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 is explosive.

I am able to assist you to the maximum. I can 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 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. He was made of wheels which when rotated was able to function similar 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.

Babbage's 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 increase their speed and the transistor and the semiconductors cause them to become smaller, more reliable and faster.

For example, ENIAC would run for 2 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 consumed 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 extreme 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 can't be able to understand them. This is 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. It is able to be aware of our surroundings, thoughts, feelings, etc. we need memory; so does the computer.

We say that memory RAM (Random Access Memory). It is the part of the computer that is similar to our awake memory. It is a sort of scratch pad where all the information that we decide to keep is kept. It is short term memory in that if the information in it is not refreshed it will be forgotten.

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 spendvaluable time and energy doing it. The computer uses a memory called ROM (Read Only Memory). Its commands that are fixed in memory that the computer needs to 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 performs 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 peripheral devices that do similar things. They connect the CPU with the outside. To communicate we talk, write, make faces, etc. We communicate with others, we write, we do things differently, etc. We can communicate with our computer usually prints on paper or puts a television screen, but, is not limited to that. Some computers can talk and understand something 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 the sell it at their cost. 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 F.T. happened to them. When you can pay your payroll, you close your doors, take your checks, and liquidate.

This is not a new story. What concerns me is that many good people, going down the drain with ideas that cannot work without capital 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 Papyrus 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. (My note: who wonders 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 was 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 came up to me and whispered so everyone could 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 items I might need from time to time. He doesn't offer magazines and books. He doesn't allow me to order 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
PROTEUS NEWS
Publisher: PROTEUS, 1190 Woodside Road, Suite 219
Redwood City, California 94062, U.S.A.
Editor: Brian Bickner, 1190 Woodside Road, Suite 219
Redwood City, California 94062, U.S.A.
Sales: Tom Kearney, 1190 Woodside Road, Suite 219
Redwood City, California 94062, U.S.A.
Subscription: All subscriptions begin at the end of the calendar year. All subscription requests should be sent to the publisher at the address listed above. Subscription rates are $12 per year, paid in advance, by check.

Copyright © 1983, PROTEUS News. All rights reserved. Reproduction in any form, in part or in full, without written permission, is prohibited. The views expressed in the articles reflect those of the author and the publisher. 180104
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 Deino) now has space to work when I'm in the office, rather than just on my off-days. I expect 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 are looking for 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 McGee 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 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 H-4.

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 half 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 overburdened 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.
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 that acts like PTOS'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. RAIDN 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:** A Basic program which will remove all REMarks from another Basic program which is stored on a PTOS disk.

**CONVRT:** This Basic program will convert ALS-8 text files to the PTOS format. This program can also convert PTOS text files back to the ALS-8 format. All line numbers will be restored.

**STORY:** 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:** An assembly language file that copies RAIDN into the assembler input stream.

**RAIDN:** (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 ($2,730,000,000,000!!) 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: A assemble language file that copies the 5 part file MEDITn in into the assembler input stream.

MEDITn: 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.

MEDITD: Documentation for MEDIT.

PAS.10: Sol 1/0 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.

STARNR: The source code for a Starwars videogame. Shoot the Imperial TIE fighters before they get you. May the Force be with you!

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

SPIN: Updated documentation for SPINWR.

SPINPD: A summary of the updated features of SPINWR: and SPIN9 found on this disk.

SPN: A device file assembled from SPINWR.

BIOPLOT: 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 PTOS 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 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: A Basic program that converts 8080 assembly language source files into TRS-80 type Z80 mnemonics. Now you can write programs for the lucrative TRS-80 market using your PTOS editor!

SIMUL: Solves simultaneous equations.

SIMLD: Saves equations in the data file below.

SIMUDATA: The data file for holding equations to be solved.
The Proteus Cassette Software Library has three new features to report. First, we have three new cassette programs which should be described elsewhere in this issue. Tape CL 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 CUPS 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 CPUG diskettes is also available on tape for $5.00, or $5.00 through the library. New CPUG 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 transferred to tape so that you can then load it into your CP/M disk system, we can handle this. Again, the price is $18 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 $18 fee.

Address orders or inquiries to:
PROTEUS CASSETTE SOFTWARE LIBRARY
C/O LEWIS MODILLER, JR.
2576 GLENDALE COURT, N.E.
CONYERS, GEORGIA 30096
Include a SASE if you want a personal reply or a library catalog.

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 programs are otherwise unobtainable.

SOFTWARE MUSIC SYNTHESIS SYSTEM—an upgrade of the old MUSIC SYSTEM with an additional voice for a 2-80 user. The overall tones are greatly improved with 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 multi-purpose 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 and does not use a card slot. The board is supplied in kit form with complete and ready 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 waveform form are inputted into each waveform. After each new waveform is inputted, the computer will play a sample scale to demonstrate the waves. The cycle of modifying and sampling can be repeated until the sound is desired. The newly configured program can then be saved by the user for future use, eliminating the necessity to have to continually reconfigure 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 $78.95 (cables not included).

Available from California Software, 207, El Cerrito, CA 94530.

(415)337-7730.
$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 contestant must certify to the origination of the invention. I.e., that it is his or her own concept and is not known in any patent or 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.
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 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 $350. 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 database structure easily. The system permits the user to add new fields, expand the size of the database, and rename fields, records, and sets. The redesign is performed dynamically, relieving the user of the chores of offloading the database, changing the structure, and then reloading the database. The user may now modify major features of the database faster and more easily.

A related product, the Dynamic Restructuring System (DRS), performs similar functions for a full, network database 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).

SOFTWARE VENDOR DIRECTORY

By MICRO-SERVE, INC.

A Reference Directory for the Micro-Computer Industry. Tells you where to go to find what you want in software and how many there are.

800 Software Vendors
32 Hardware Vendors
105 Software Categories

Also available on disk - write for more details

Now in its third printing
Now offering a Subscription Update Service

ORDER TODAY!

Directory $57.95
Subscription Update Service $25 for 6 months

- operating systems
  - General Business
- languages
  - Micro-Serve, Inc.
- utilities
  - Applications by
- data base
  - Nyack, NY 10960
  - and many more
- managers
  - (914) 358-1340
  - 105 in total
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 jumped to come up in the "protected" mode after a power failure. The difficulty is that the Sol has not used pins 20 and 70 of the 8-100 bus 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 circuits (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 8-100 bus 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

[Diagram showing connections and inverter]

NOTES: U44 has a spare inverter which may be used. On an FC Solo, J11 pin 70 may be grounded too. This ground must be out.

ONE PORT AND TWO PRINTERS

John Corman 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 John'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 circuitry 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 amazing performance since both devices can be on line at the same time. Imagine a printer printing out some text while 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?

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 separate drivers for the two printers since the Sol only "sees" one of them at a time.

QUBE 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 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 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 D" 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...
Some programs will crash if you try to RESET them out of an infinite status (my Spinnwriter, I set contact 5 of SW 1 to "ON" (up) on the control panel circuit board (C98BHF). 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 19 to pin 8. If this program will be used with PDTSO it must be preceded by a driver table. See Section 9 of the PDTSO manual for an example of a PDTSO driver table. If the printer was given in Sol's manual the driver was never changed. 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 on a disk on his WW system disk called PRINTER. He can copy it over to his BASIC disk by using the PDTSO "GET" command. Put the WW system disk in drive 1 and the BASIC disk in drive 0. Then type after the PDTSO 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 OP=#1 (*#1 is the screen driver in PDTSO)

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 1/0, 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 3X 19 C9 OUTPUT LDA FLAG Get the print flag
C903 FE FF CC CC CC C9 CFI OPFF Test print flag
C907 21 CC C2 PRINT If match, print
C908 AF 21 8A XRA A Get pseudo port for VDM
C909 C3 1C 00 JMP 0C01CH Solos output routine

C90C DB F8 PRINT IN OPBF Get status word
C90E E6 A0 ANI 0AFH Mask off all but URT and CTS bits
C910 9C 00 CFI OPAD Test for both bits present
C912 C2 0C 9F JNZ PRINT If missing, keep looping
C915 06 FF MOV A,B Put character in A
C916 D3 F9 OUT OPBF Send it out
C918 9C RET Return
C919 00 FLG DB 0 Store initial 0 for no print

Notes:
1. In order to have selective printing, a flag byte is stored at location C919H (51461 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 receives the Sol's data until the end of the command. When you press the printer it should be the same as the print port and the printer sends a CTS signal when it is ready. The Sol's manual internal circuitry which "pulls up" the status bits to a positive value when printing is connected to the printer's "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.

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 LF but only carriage returns. I seem to recall that Michael Shryer published some notes on how to replace all those LFAs (OAH) in the program with CRs (ODH). Failing to find that info, I suggest that you search the program code for OAH's and selectively change them to ODH'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 search/replace 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, Centronics, Spinwriter and Selectric device drivers for use with WordWizard.

Answer: Drivers for all of these printers are available on either the Diablo Diskette (available through Proteus) or 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 to the WordWizard inforamtion for the printer in the app-averager 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 PDTSO "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 assembled 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 particular good examples of format and clarity in the Vol. 3, #5, 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 think it's also true that many of the amateurs' problems can be found in previous issues of Proteus.

The technique you read this, Northstar should have its new, single board computer out to cash in on the TRS-80 market. For those Sol/Northstar users, there is a new DOS, BASIC and Utilities. It is called version 5.2 (S) or (DG) and is COMPLETELY RECODABLE. Also, some of the new Utilities (PANTEST) are written in 808 code.
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 288 system because of timing problems. (at 4 MHz) I know from comments I have read in the newsvetle 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 Maquire
P.O. Box 3742 DT
Anchorage AK, 99510

A Report on Northstar Computers’ Hard Disk System

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

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 288 CPU (the HDOS is written in 288 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 8800 CPU based system. The HDOS (hard disk operating system) occupies the first 18K of memory beginning from address 100 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 EX (execute byte), EM (examine memory), EP (examine port) and so on. It is obvious why 288 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 8800 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 a Horizon so I was able to make use of the system. The drive is a Century Data Marksmen which is contained in a sealed unit. The capacity of 18 megsobytes allowed me to transfer my 50 minidisks of programs and data over to the HD with still some more room to go. This new found convenience is now without 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 30 to 48 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 FOO

Relocating SOLOS to FOO (PROTRES News, V.2, #3, p.2) seems like an ideal way to expand into a CP/M system larger than the 48K allowed with SOLAS at C000. Bob Goodman's hardware instructions were easy enough to follow (jumper pins 9 & 12 of U23, pull pin I of U24 to +v through a 1.5K resistor, but I did have some difficulty relocating SOLAS. The source code from the CP/M User's Group had been modified slightly from the published listing of SOLAS. Since some programs published in PROTRES call obscure routines in SOLAS, I returned the source code back to its original form to maintain address compatibility, except for the high order byte. Changes made to relocate SOLAS included changing the origin from C000 to F000, changing the TM address space to F000, and changing the start of system RAM to F000.

A less obvious change that was necessary was in the 'clear screen' routine, KB41 (C00A-C009). SOLAS assumes that the VIM is at C000-FFFF. It checks for the end of the screen by comparing the high order byte of the current cursor position to 00 (the first address after CFFFF). If you move the VIM to F000, this routine fails; in fact, the 'clear screen' function doesn't work even if you change '00' to '01'. It fails on the instruction 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 JNC (jump on zero).

For the adventurous: I have sent a copy of the relocated SOLAS source code (under CP/E and 2) the object code of same (on OATS tape) to our program librarian, Lewis Howley.

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 SOLAS' project. Said project is a rewritten, expanded SOLAS relocated to F000 on a 'smart' Personality Module, i.e., the modification will allow you to switch SOLAS at C000 and at F000 simply by switching ROMs, no external switching required. Watch PROTRES for further details, or contact Lee at Colenos, 2695 5th Street, Berkeley, CA 94710.

Now - HELP! How do I go about changing all my Prottech software so that it runs with the VIM at F000?

Bob Steck, 19 Mayfield Road, Regina, Saskatchewan Canada

ANOTHER NOTE ON PSCAN.

by Graevson Evans

Previous articles that have dealt with the PSCAN routine in PTBUG 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 PTBUG 1.1 the input command line is stored in a buffer at 9FF0H. 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 RSBUD system call to return from your program to PTBUG 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 Thinks 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.

EQUIPMENT REVIEW

Memory boards and 280 processors.

I have some experience with a few memory boards that I have not seen reviewed in PROTEUS. Since I am running P200 with the Helios, memory boards have been a pain in the whatever.

After trying at least eight 16K and 32K boards in my system to find a few good boards I have a big list of boards that don't work and a list of boards that do. Some you have already read about.

Calf Computer Systems K21 16K static board mentioned. It is loaded with switches for selecting this and that bank select options, addressing 4K blocks, etc. It has LEDs to indicate board select and board active. Manual is very good.

The old Altair 88-16K memory board also works fine. It is present in the classifieds at reasonable prices. This is a 16K static board and can be addressed at 16K increments.

I have not had good luck with Goddard Econorams, Cromemco's 16K bank select boards, or E.E.T 8K static boards.

For those of you who would like to run a 280 processor board in something other than a Sol using the Helios drive I would recommend Cromemco's 280 CPU board (NOT the single board computer which don't work). It is the ONLY 280 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 8810 signals including interrupt enable.

.Grayson Evans
1243 Manchester Ave.
Norfolk, Virginia 23508

MINOR IMPROVEMENTS 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 overheated resistors and a high density of IC's. The heat dissipation capabilities can be improved by making the following modifications:

On the Controller board install extra heat sink to area to the 5V regulators by slipping (with epoxy) Themalloy THM6010 heat sinks to the top (plastic) of the regulators. Apply a thin coat of 50 ohm resistors to area across the regulator, until the epoxy is dry. They will fit nicely within the present heat sink. Other heat sinks could work just as long as the total height is less than 3/4".

Also, since the Fairchild 9403 FIFO chips run very hot, apply a Themalloy THM6002B to the top of these chips in a similar manner. These extensions are designed to be bonded to the top of chips and attached to the board's power distribution network. These allow for the assembly of the illustrated circuit board and header. The 2M 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 one. This must be done using a vacuum powered solder sucker. Do not attempt to remove these connectors using a solder wick or the various sprays 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 their place smaller size and larger value caps. I recommend capacitors such as Mallory C1CP04MSU10C at about $0.19 for 50 or S&I-Peave 2C25U104X705C4 at about $.24 for 50. These capacitors are small (don't stick up above the chips), .1uf 50V ceramic caps are designed for bypass. 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 Eisten

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 but 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 4 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 single 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 with 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 14. Used to enable the external gates as well as those on the CUTS board.
4. Write clock out, S-100 pin 58. 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 KiloHaud 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 KiloHaud.
INTERFACING A SOL COMPUTER TO AN H-14 PRINTER

By Bill Surveyk

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).  

<table>
<thead>
<tr>
<th>H-14</th>
<th>SOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGN</td>
<td>1</td>
</tr>
<tr>
<td>OUT</td>
<td>3</td>
</tr>
<tr>
<td>SIN</td>
<td>5</td>
</tr>
<tr>
<td>RTS</td>
<td>7</td>
</tr>
<tr>
<td>CTS</td>
<td>9</td>
</tr>
<tr>
<td>DSR</td>
<td>11</td>
</tr>
<tr>
<td>SND</td>
<td>13</td>
</tr>
<tr>
<td>DTR</td>
<td>14</td>
</tr>
<tr>
<td>FGN</td>
<td>1</td>
</tr>
<tr>
<td>OUT</td>
<td>3</td>
</tr>
<tr>
<td>SIN</td>
<td>5</td>
</tr>
<tr>
<td>RTS</td>
<td>7</td>
</tr>
<tr>
<td>CTS</td>
<td>9</td>
</tr>
<tr>
<td>DSR</td>
<td>11</td>
</tr>
<tr>
<td>SND</td>
<td>13</td>
</tr>
<tr>
<td>DTR</td>
<td>14</td>
</tr>
</tbody>
</table>

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 facing up at you through their rectangular holes. Using a pointed tool, depress, on the left side of shiny part, the tiny little rectangular tabs holding the female pins in the square holes and simultaneously outside 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 0=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 NOT PARITY, 2 STOP BITS, and 8-BIT DATA WORD LENGTH).

SET 0=1  <CR>
DU 00 0000  <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 than 110 is to be used, then handshaking must be employed. Also, the characters being printed by the H-14 will not be displayed on the monitor unless a special routine is written and included in the handshake program. (see Figure 2).

| CB00 | DB FR 0 | 0001 | HANDS IN PRTST | J set status byte | get status byte
| CB02 | DB 20 | 0002 | ANI MASK | J set status bit | (1=Free)
| CB04 | CA 00 | 0003 | JE HANDS | J go back if busy
| CB07 | JE 81 | 0004 | JU R | J select serial
| CB09 | CD 1C | 0005 | CALL ROUT | J output character
| CB0C | AF | 0006 | XRA R | J screen (R=0)
| CB0D | C3 1C | 0007 | JMP ROUT | J output character
| CB10 | 0005 | NOTE: uses ROUT
| CB10 | PRTST | EQU | 0FSH | J SET RETURN
| CB10 | 0010 | MASK | EQU | 020H | J SOLOS status
| CB10 | 0011 | ROUT | EQU | 0C0ICH | J PORT
| CB10 | 0005 | SOLOS OUTPUT | J bit 6 is the
| CB10 | 0007 | 0005 | 0007 | J status port
| CB10 | HANDS | 0003 | 0002 | J
| CB10 | PRTST | 00FS | 0001 | J routine

Figure 2: Handshake and Display Program

To enter the Handshake and Display Program go into SOLOS and type in the following:

```c
CB00 DB FS BE 20 CA 00 CB 3E 01 CD 1C 00 AF C3 1C 00/<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:

```c
GET PRINT <CR>
```

```c
SET ROUT=CB00 <CR>
```

In BASIC (Pro Tech Extended) use the SET OP=3 command to start the H-14 printing and SET OP=0 to stop.

In SOLOS use 0=3 and 0=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 NOT PARITY, 1 STOP BIT, and 8-BIT DATA WORD LENGTH.

One of the features of the H-14 is the defining or changing of the width of characters (and thereby the size) and the number of lines of characters per inch (also changing the size). This can be done in BASIC by using the PRINT command incorporating the ESCAPE key character and a specified letter within quotation marks, such as

```c
100 PRINT "&Cw"
```

or

```c
100 PRINT &Cw
```

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

```c
100 PRINT CHR(12)
```

The character control codes are in the H-14 Operations manual for your reference.

Bill Surveyk
8014-163 St., Edmonton
Alta., Canada T6N 2H3

Happy printing!!
DEVICE DRIVERS EXPLAINED
Grayson Evans
1245 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 an overview of their purpose, driver design and a detailed review of use of entry points and two examples of simple device drivers—ones that do nothing and one that does output.

The concept of device drivers is not new. Anyone familiar with the PDP-1 computer (a small computer designed to run on DEC computers) will be familiar with this use. I don't 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 consistent 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 (input/output ports) to "look" to the system and user written software is provided to do the same privileges as, the mass storage file system. In the case of PTOS 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. The use of 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 PTOS is a good one but unfortunately a good explanation of how to take advantage of it is not available and how it works is not explained in the manual. After a brief introduction and entry point explanation in Section 3 of the PTOS manual, the user is left to fend for himself with the eye-staining Cutter example found in Appendix C. Information about device driver operation can be found in Section 3.6, scattered throughout system calls in Section 2, Appendix A, and Appendix C.

The information is this article has been gathered from the PTOS manual, trial and error, experimentation, calls to PTC, and debugging 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 PTOS and an I/O device attached to the system. Since PTOS cannot be written to anticipate all the gadgets attached 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 PTOS and handles the transfer of data to and 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 operates the program on the device file. Since the device is now treated as a disc file, the file must first be opened. This is where PTOS

...overs 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. PTOS is responsible for finding out whether to transfer the data to or from a track on the disk or to and from the new 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 device. The first 25 bytes of the program are called 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 I/O 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 or disk. It is a front panel light or a random access disk file. The table is compiled as a set of addresses. The addresses are used by PTOS 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 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 PTOS manual gives an explanation of the driver table organization. Notice from the table on the page that the subroutines allow the device driver to perform (possible returns back to PTOS. These are called the CALL41 return and the CALL42 return. The CALL41 return refers to a normal RET instruction return to the first byte after the last CALL instruction. A CALL42 return refers to a return to the forth byte after the last CALL instruction "performed by firing the handler on the top-of-stack and then performing a RET...". If an operation has two possible returns then by PTOS convention the first return is the first return is the first return. If some error or abnormal condition occurs (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 INITIALIZE operations only use one (CALL41) 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 (RBLOAD). Using the file number supplied in the call, PTOS knows the read is directed to a loaded device driver. PTOS obtains the address of the code to perform the operation from the READ BLOCK entry (DRBR) 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 PTOS thru the CALL42 instruction. PTOS then returns control to the calling program passing information about the operation in various registers. If PTOS found a value of zero for the address of the read operation in the driver table, it would indicate that the device driver was not capable of performing a read operation and PTOS would issue an "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 I/O. These are not addressed but are
The DTINI entry is automatically called by PTDS 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 system manual, these examples have been made 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 PTDS manual the DTWBR entry is called whenever a write is made to the device. The DTWBR entry accepts a byte and returns the number of bytes written in the HL register pair.

The second routine is one that would be used by the driver for cases where the executing program may ask for the device's status. The DTCLO operation sets up the serial port interface. Of the operations provided by the device, only two are required: CLOSE and INITIALIZE. The DTINI entry point must have an address since this entry is taken automatically by PTDS when the file is opened. The DTCLO entry must be provided so PTDS can close the file otherwise it will remain open until PTDS is reloaded. 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 written by a program using this driver is printed immediately. Actually, with a block size of one, the value of ITO doesn't matter. The results will be the same. Notice how the INIT 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 that is greater than one. The output buffer is not empty when the drivers' write routine is called. The executing program executes and END-FILE system call to the file, PTDS will call the drivers write routine to write what's in the buffer and the buffer will not be emptied (reset the buffer pointer the start of the buffer).

A subsequent CLOSE system call by the executing program will cause PTDS to write the contents of the buffer (by the drivers DTWBR 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 from 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 DTBR and DTBNB entry points. An entry for DTCLO and DTFP 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 PTDS to the driver, DE contains...
the number of bytes to read. If an EOF condition is detected by the read routine before all 8 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 with HL=x000. If an EOF return is taken by the driver after reading a number of bytes less than 8 on a particular call then those bytes will be lost even though HL contains 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 compiled 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 beginning of the CUTOFF 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,
```

Its now ready to go. Don't 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 analog tape board, a time stamped 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
  ; CMDN:  EQU 50H
  ; REG:   EQU 51H
  ; TREG:  EQU 51H
  ; CONOUT: EQU 0D89FH
  ; copy equ
  ;
  ; org 8000H ;Good neighborhood to stick drivers
  **********
  DTRD DW D
  DTRNB DW 0
  DTRLB DW 0
  DTWR DW WRITE ;Write buffer entry
  DTBW DW WRITE
  DREW DW 0
  DTEOF DW EOFWT
  DTCLO DW CLOSE
  DTWEL DW 0
  CTWEL DW 0
  DBLK DW DB 1 ;only one byte "BLOCK"
  DTITDW DB 1 ;immediate transfer on
  DTINIC DW INIT
  **********
  * INITIALIZE the serial port on the UART board
  * INIT:  MVI A,1
  OUT 52H
  XRA A
  OUT 53H ;disable interrupts
  MVI A,04 ;set baud rate to 300
  OUT 98H
  CMND
  RET
  * WRITE - Entry point for writing one block.
  * WRITE:  MOV A,0 ;write DE>0
  OUT 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.
  * EOFWT:  MOV A,0 ;same routine as WRITE
  OUT E
  JZ EOFEXIT
  MOV A,M
  CALL PRINT
  INX H
  DCX D
  JMP EOFWT
  * Prints one character on printer
  * PRINT:  PUSH PSW
  PRINT:  IN ANI 80H
  JZ PRINT1
  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 LP'S
  CLO1:  MVI A,LF
  CALL PRINT
  BCR 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
R2
CALL CONOUT
INX H
JMP MSG

ENDMSG: ASC "PRINTING COMPLETE"
DB BELL,BELL,BELL,CR,LF,0

END

********************************************************************************
* PTDVRDR - Driver for reading from paper tape
* reader.
********************************************************************************

ORG 8000H
;Good place to stick drivers

DTRB DW 0
DTRN DR READ ;read a block
DTRLB DW READ ;read next block
DTWRR DW 0
DTWB DW 0
DTREW DW 0
DTEOF DW EOFIT ;EOF entry
DTCLO DW CLOIT ;Close entry
DTSK DW 0

DTRBLK DW 32 ;"block" size
DTRITK DW 0 ;meaningless for read
DTRITK DW INIT ;initialize

INIT - Initialization entry

INIT: XRA A
OUT 20H
OUT 21H
OUT 22H
MV B,A,OFFH
OUT 23H
MV A,2EH
OUT 20H
MV A,3EH
OUT 22H
OUT 21H
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: MV C,D
;C = number of bytes read
IN OFFH ;check for "EOF" condition
ORA A
JNZ EOFEXIT

READ1: MOV A,D ;test for zero in DE

OR A E
;which indicates read done
JZ DONE

CALL GETBYTE ;get a byte from reader
JC DONE ;"EOF" detected during read
MOV M,A
INX H
INR H
INR C
DCX D ;decrement bytes to read.
JMP READ1

GETBYTE: IN 20H ;reader byte ready?
ANI 80H
JNZ GETB1
IN OFFH ;check for "EOF"
ORA A
JZ GETBYTE ;no EOF, keep trying
STC
RET
GETB1: IN 21H
RET

EXIT ROUTINES

DONE: XTBL ;normal CALL+2 return
INX H
INX H ;all DE bytes were read or some
INX H ;bytes were read prior to EOF.
INX H
XTBL
EOFEXIT: MV A,2EH ;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 PTDOCS manual for version 1.4. The new manual for PTDOCS
1.5 has different numbers.]
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 80 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,

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. Mc Kelve y
* 330 S. State Street
* Ann Arbor, MI 48104
* COPY NPTDEPS
  ORG 0D60H
  XEQ 0D60H

******************************************************************************

*** MODE 1 INITIALIZATION & COMMAND LOOP ***

INITIALIZE MCP & GO TO MODE 1

| STR     | LXi SP, STK | set stack pointer |
| CALL    | LXi H,SON | set msg ptr |
| CALL    | LXi WRT | write it |
| LXi TTP | set text pointer |
| MVX A,CRD | set default CR |
| STA DCR | store it |
| STA AUT | set default auto-send |
| MVX A,APS | store it |
| MVX A,DPB | set default dupl mode |
| STA DTM | store it |
| STA DMA | set default receive mode char |
| MVX A,DISP | store it |
| STA BRT | get high speed bit |
| MVX A,WHI | switch modem |
* MODE 1 COMMAND LOOP

** CALL CRP **
write a LF/CR

** CALL CRF **
and another

** PUSH H **
save the text pointer

** LXI H,MMU **
load the menu addr

** MOV H,255 **
load the counter

** CALL L1 **
WRT w/ bigger count

** POP H **
get the text pointer

** CALL CRP **
write a LF/CR

** CALL CRF **
and another

** CALL CCF **
get console char

** ANI 5PH **
conv. to UC/NP

** CPI 1BH **
is it an 'ESCAPE'?

** JZ EXIT **
if yes, exit to PTOS

** 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 D7

** JZ CAS **
if yes, chg A/S

** CPI 43H **
is it a C7

** JZ ACR **
if yes, adj BOL char

** CPI 46H **
is it an F?

** JZ DPX **
if yes, DPX adj

** CPI 57H **
is it a W7

** JZ RMM **
if yes, RCV M. set

** CPI 5AH **
is it a B7

** JZ DCC **
if yes, comm display

** CPI 51H **
is it a O7

** JZ STK **
if yes, reinitialize

** CPI 4FH **
is it an O7

** JZ KDFILE **
if yes, load buffer with outfile

** CPI 42H **
is it a B7

** JZ ABR **
if yes, adjust baud rate

** CALL SWP **
write char

** CALL INV **
write error msg

** JMP LOP **
loop again

** D1 STA DPL **
store it

** JMP LOP **
go to MODE 1

** D2 STA DPL **
store it

** MVP A,41H **
load an A in A

** JMP LOP **
go to MODE 1

* 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,65A **
load text addr

** CALL WRT **
write the string

** POP H **
get the text pointer

** CALL CCF **
get console char

** ANI 5PH **
conv. to UC/NP

** CPI 1BH **
is it a D7

** JEE A1 **
else a C7

** MVP A,13H **
put a DC-3 in A

** STA DCR **
store it

** JMP LOP **
go to MODE 1

** A1 CPI 43H **
is it a C7?

** JEE A2 **
else, try again

** MVP A,06H **
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.

** DFX **

** 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 console char

** ANI 5PH **
conv. to UC/NP

** CALL SWP **
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 DFX **
and try again

** D1 STA DPL **
store it

** JMP LOP **
go to MODE 1

* STORE THE FOX COMMAND AND SET SEND MODE BYTE
* TO 'AUTO-SEND'.

** D2 STA DPL **
store it

** MVP 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 5PH **
conv. to UC/NP

** CALL SWP **
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 an error msg

** JMP RMM **
and try again

** M1 STA DMA **
store it

** JMP LOP **
go to MODE 1

** 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

** 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

17
* 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 WRW  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
  LXI H,20 load the counter
  MOV A,M ready to write
  INC H increment the pointer
  DCR B decrement the counter
  PUSH B save count
  CALL COT write it
  POP B restore count
  CALL COT write it
  MOV A,B ready to count
  CPI 0 is it 0 yet?
  JNE L1 else, try again
  RET . else, continue

*******************************************************************************

** MODE 2 COMMAND LOOP ***

* ISSUE A PROMPT AND GO TO MODE 2

SNDD CALL ERA erase screen
  LXI H,33H load prompt in A
  CALL COT write it
  CALL CRF write a LF/CR

* THE MODEM INPUT LOOP

SNDD 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 SNDD check again

* THE MODE 2 COMMAND LOOP

S1 CALL CSCC check the console status
  JS SISD if nothing, check modem
  CALL CCF get char
  ANI 7FH and off the parity
  CPI 1BH is it an 'ESC'? or 'CTRL-A'?
  JS LOP go to MODE 1
  CPI L2H is it a 'CTRL-B'?
  JS RCVT go to MODE 3
  PUSH PSW save A & FLAGS

S2 IN MST get the modem status
  RAL 1 put high bit in carry
  JNC S2 loop until ready
  CALL CCF get char
  CALL CRF and off the parity
  CPI 1BH is it a 'CTRL-C'?
  JS XMT send some text
  CPI 0L is it a 'CTRL-A'?
  JS ISD call LOGON #1
  CPI 02H is it a 'CTRL-B'?
*: SEND ON EOL CUE.
*: 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.
*: THE BASIC TRANSMIT LOOP: MODEM AND CONSOLE STATUS
*: BYTES ARE CHECKED TO PROVIDE EXITS WHEN NECESSARY.
*: SEND A DC-3 OR A CR, THEN OBEY ATS COMMAND.
*: DECIDE IF OPTION IS CONTINUOUS SEND, SEND ON
*: COMMAND, OR SEND ON CUE
*: CALL LOGON #2
*: call LOGON #2
*: CPI ODH
*: if is a CR?
*: CS DEF
*: if yes, call EOL byte
*: OUT MOD
*: send console char
*: CALL SPC
*: write it
*: JMP SNL
*: check modem again
*: XCT
*: 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
*: Y1
*: get modem status byte
*: IN MST
*: get input status bit
*: ANI MSM
*: nothing? try console
*: JZ Y2
*: if yes, get the input byte
*: ANI 7PH
*: and off the parity
*: CALL SPW
*: write it
*: JMP Y1
*: check again
*: Y2
*: check console status
*: CALL CSC
*: if data, check it
*: JMP Y1
*: get char
*: MOV A,M
*: increment the pointer
*: CPI OFFH
*: is it a pad char?
*: JZ EOT
*: yes? write msg & exit
*: CPI ODH
*: 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.
*: C3
*: CALL CSC
*: a way out if the
*: JMP SNC
*: 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 7PH
*: and off parity
*: CALL SPW
*: write it
*: CPI ODH
*: 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,4008H
*: load the counter
*: DCX B
*: decrement the counter
*: CPI ODH
*: get modem status byte
*: JMP C6
*: if data, write it
*: ANI MSM
*: save the status bit
*: JNZ C5
*: if data, write it
*: CPI 0
*: is it 0 yet?
*: JNZ C5
*: else, loop again
*: JMP Y1
*: yes? send next line
*: C5
*: IN MST
*: get input byte
*: ANI 7PH
*: 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 CRP
*: write a LF/CR
*: CALL CRP
*: one more, please
*: LXI H,18H
*: load msg address
*: CALL WRK
*: write the string
*: CALL CRP
*: write a LF/CR
*: POP H
*: get pointer
*: INX H
*: increment H
*: JMP SNL
*: go to MODE 2
*: AUTOMATIC LOGON MESSAGES $1 & $2
*: THE MSG $1 ENTRY POINT
*: ISO
*: push H
*: save the text pointer
*: LXI H,19H
*: load the text address
*: CALL 11
*: send it
*: POP H
*: get pointer
*: JMP SNL
*: go to MODE 2
*: THE MSG $2 ENTRY POINT
*: T5O
*: push H
*: save the text pointer
*: LXI H,16H
*: load the text address
*: CALL 11
*: send it
*: POP H
*: get pointer
*: JMP SNL
*: go to MODE 2
*: 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
DEC R decrement the counter
OUT MOD send the char
CALL SPC write it
MOV A,B ready to count
JNZ 0 is it 0 yet?
CPI 0 is it 0 yet?
JNZ I2 else, try again
RTS head for MODE 2

******************************************************************************

*** MODE 3 COMMAND LOOP ***

* WRITE A PROMPT AND ENTER MODE 3.

RCVT CALL ERA erase screen
CALL COT load prompt in A
CALL CRP write a LF/CR

*** THE MODE 3 COMMAND LOOP ***

MVI M,OFFH put text pad in buffer
INX H increment the pointer
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 DDA 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 00H 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
POP PSW get the char
R3
MOV M,A put it in the buffer
CALL SPW write it

R2 CALL CSC get console status
JZ R1 nothing? check modem
CALL CCP fetch console character
ANI 7FH and off parity
CPI 1BH is it an 'ESCAPE'?
JZ R6 if yes, go to MODE 2

R5 CALL SPC write it
PUSH PSW save A and flags
INM R5 get the modem output status
RAL. put high bit in carry
JNC R5 loop if not ready
POH PSW get A & flags
CPI 0DH is it a CRT?
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,OFFH insert buffer pad char
INX H increment pointer
JMP SNOT 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 4BH is it an HT?
JNZ P1 else, go to timer
POH PSW get char
JMP SPW write it

P1 IN MST get modem status byte
ANI LSM save status bit
JNZ P2 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

CRP PUSH PSW save A & flags
LDA DPL put CR in A
CALL COT write it
MVI A,0AH put LF in A
MVI 0AH put LF in A

* ERASE SCREEN

ERA PUSH PSW save A & flags
MVI B,16 load counter
R1 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
CFI 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
* MODEN 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,OPFILE set pointer to filename stg
         CALL CCP  get console char
         CPI 0DH  is it a CR?
         JZ F2   if yes, continue
         CALL SPW  write it
         MOV A,H  store it
         INX H   increment the pointer
         JMP F1   go back for next char
         LXI D,OPFILE load D with filename pointer
         LXI H,0   load H with buffering parameter
         CALL SYS  call to system
         DB OPPOP  open file
         JMP RDFILE  if error, try again
         LXI H,TPT  load H with text pointer
         MVI H,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 KBLOP  read file into buffer
         NOP
         NOP
         POP PSW  get A & flags
         CALL SYS  call to system
         DB CLOOP  close file
         NOP
         NOP
         NOP
         NOP
         POP PSW  get A & flags
         CALL SYS  call to system
         DB GLOP  close file

         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 CCP  get console char
       ANI 5FH  conv. to UC/NP
       CPI 5FH  is it a Y?
       JNZ PTROS  no?, return to PTROS
       LXI D,FNAME load D with file name pointer
       LXI H,0   load H with buffering parameter
       CALL SYS  call to system
       DB OPPOP  open file
       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
       CALL SYS  call to system
       DB EOPOP  end file
       NOP
       NOP
       CALL SYS  call to system
       DB DLOOP  close file
       NOP
       NOP
       PTROS    CALL SYS  call to system
                 DB RETOP  return toPTROS

******************************************************************************

* *** 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

21
**M. MCKELVEY CONTINUED FROM PAGE 21**

```
DB OAH
ASC 'C'  EOL CHARACTER  DC-3'
DB O9H
DB OAH
ASC 'F'  DUPLEX MODE  HALF'
DB ODH
DB OAH
ASC 'B'  BAUD RATE  1200'
DB ODH
DB OAH
ASC 'O'  LOAD OUT FILE'
DB ODH
DB OAH
ASC 'O'  REINITIALIZE'
DB ODH
DB OAH
ASC 'E'  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 'DUPLAX MODE (P,H):'
RMT  ASC 'RECEIVE MODE (N,T):'
SPD  ASC 'BAUD RATE (L,H):'
ETM  ASC **** BOT ***
FSE  ASC 'press any key'
NAME  ASC 'OUT FILE NAME?'
QFILE  ASC '0000000000'
FNAME  ASC 'MP3DATA'
LOGON MESSAGE $1
IST  ASC ''
DB ODH
DB OAH
ASC ''
DB 13H
LOGON MESSAGE $2
TST  ASC ' '
DB 13H

***************************************************************************
*** GLOBAL EQUATES ***

CSC EQU 001H  Cons Status Check (active low)
CSF EQU 002H  Cons Chac Fetch
COT EQU 003H  Cons Output
MOD EQU 005H  Modem data port
MST EQU 008H  Modem Status port
SDR EQU 00AH  Speed relay port
SHI EQU 00H  Sol HI speed bit
SLO EQU 08H  Sol LO speed bit

***************************************************************************

**END OF PROGRAM**

***************************************************************************

**UNCLASSIFIED ADS**

**COMPUTER PORT**

WACKER/DRP/INDEX/SELECT
FDroid UTILITIES
MAILBOX
JEGUIGY SYSTEM
TEXT EDITOR
DISASM/ASMBLB II

$150.00  65.00  45.00  160.00  75.00  40.00

FOR SALE:
1. SOL 20 with 48K plus monitor and cassette
1. CP-80A paper reader
1. Micropolis 5 1/2" 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 Kittel
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 651
model 110, B" floppy disk drives with documentation. These units are completely
compatible and interchangeable with Shugart 800/801 drives.

Edward Endler
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;
asser, games...In excellent condition. Price $1000.00 or
best offer.

Mario I. Guerrero
1021 Nw 84th Dr.
Coral Springs, FL 33065
H 305-752-6737  W 305-587-2900 EX6225
```
I might be interested in an empty SOL if I could locate 48-2108 chips, NM type, for my extra Extensa RAM board. I have dual Micropolis Mod II drives and two controllers and would appreciate your comments on any possibilities and advantages of using a second SOL with them independently and in combination.

Charles L. Hunsing
4741 Hidalgo Ave.
Middletown, PA 17057
(610) 926-1177

I HAVE THREE PIECES OF SOFTWARE ON EBCONIC THAT I USE A LOT AND WILL SHARE, SHARP WITH OTHERS WHO WILL SEND REASONABLE PROGRAMS IN EXCHANGE NO GAMES CARDS...BUT FINANCE, ANALYSIS, SOME HELP CONVERTING MY EBCONIC TO DSK BASIC, ETC., WILL BRING A DELIGHTFUL PROGRAM FOR SOL-20 QUITERS WITH CUTS AND NO DISK.

WHAT I WANT IS A DATA BASE PROGRAM WITH FULL SORT, REPORT AND ABILITY TO MATHEMATICALLY INTERRELATE COLUMNS...A HOUSEHOLD INVENTORY AND INSURANCE PROGRAM THAT WONT SORT BUT WILL GREATLY HELP INSURE AND KEEP TRACK OF ALL YOUR GOODIES AND A MULTIPLE REGRESSION PROGRAM THAT WILL RUN, RECORDING PATH FROM TAPE OR RAM, ANY NUMBER OF VARIABLES AND REGRESSIONS THAT YOUR RAM WILL ALLOW; AND GIVE A FULL PRINT OUT OF RESULT

George C. Hammer
9228 Embassy Bangkok Keg
San Francisco, CA 94134

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 Pellan 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) 924-5038.

WANTED: PCT's disk FORTRAN for the Helios and/or WordWizard. 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 10, $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
Breckwood Square 11661 San Vicente, Suite 603, Los Angeles, 90049 (213) 820-2096
San Francisco Penthouse Telephone (415) 746-3811

Distributo
3331 Union City Blvd. - Union City, CA 94587 - (415) 488-1850

Foodservice Computer Systems, Inc.

Distributo/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 backdrop, self-correction, and software for a North Star drive.

My asking price is $1,675.

Joe Lancaster
191 Cedar Ridge Drive, $10
Stockton, CA 95207
Phone (home): (209) 957-7618

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 pick up in either San Francisco or Los Angeles area. Joe Huspite, 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 Bjortand
P.O. Box 13172
Oakland, CA 94661
MODIFICATION OF CP/M

1) On CONIN: Input, the Console Command Processor flags
CTRL-U and CTRL-X as line delete characters. In v. 1.14 of CP/M
(435 double density, the poll for these characters begins at
1AFH above the base of BDOS (130FH in the SYGSEEN position).
To substitute & for CTRL-X as a line delete key, make the following
patch:

Original code:
AAA9 FE 18 CPI 18H ;CTRL-X ???
Modification:
AAA9 FE 40 CPI 40H ;& ???

2) To implement those special features used only during
CONSOLE: Output, modification of the CP/M "BEAD" 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 6DFH above the base of BDOS (130FH in the
SYGSEEN 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:
AA9FH FE 7F CPI 7FH ;DEL (rubout) ???
AA9F9 FE 40 CPI 40H ;JZ AB9FH
AA9FH CD 18 BB CALL 8B19 ;CONCTRL

ADDENDUM

1) The SOLUSERO assembly listing was generated using the
Northstar (TSSD) PASCAL 3880 assembler.
2) Polling routine for Anderson/Jacobson 811I 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-1a (note -- the protective ground, pin 1, is not used by the
A/J). To operate the A/J at 9600 baud, configure its output
speed to 9600 baud. Use the following section of the SOLUSERO
Listing: routine prior to
CALL SDOTT.

SERST .EQU 0FBH ;Serial Status Port
SDSR .EQU 2 ;Serial Data Set Ready
WAIT: IN SERST ;is A/J 811I buffer full?
ANI SDSR ;JNZ WAIT ;if so, wait until empty.

3) Did you buy a copy of the ELECTRIC PENCIL (v. SS-11A for
435 double density) to use with the SOL and a selectric printer only to
find that selection of single-spacing (LINE SPACING S1) in the
PRINT SUB-SYSTEM COMMAND TABLE resulted in a printed text
with double-spacing? S1 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>
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 OA 0A
1133H 0D 0D

4) The DEL protocol in the CONOUT: routine of SOLUSER9 will
fail if you attempt to delete a backlash, as do all of the other
published corrections for this problem. With CP/M, delete the
entire command line with Ø. When using either CP/M or MSASIC, if
you inadvertently attempt to delete a backlash, return control
to the command mode by typing a RETURN.

SOLUSER9 FILE: SYSTEM.WRK.TEXT

00001 .PROC SOLUSER9
00002 Current memory available: 5088
00003 ; * N+S CP/M User Area Routines for SOL/20 *
00004 ; * System constants *
00005 ; MSIZE .EQU 46 ;Decimal Size of System
00006 ; BBS .EQU 0A0DH
00007 ; USER .EQU 0B80DH
00008 ; CPRINTF .EQU 0B80D8 ;Sets buffer location
00009 ; ORG 0B80D8
0000A BB01 ; * Define Constants *
0000B BB02 .SOLJPT .EQU $ ;SOLUSER Jump Table
0000C BB03 .SOLOS .EQU $0C04H ;SOLOS Warm Entry Point
0000D BB04 .KSTAT .EQU $02E ;Keyboard Entry Driver
0000E BB05 .SDRT .EQU $0C0AH ;Serial Data Output
0000F BB06 .VIDMOT .EQU $0C59H ;Video Display Routines
00010 BB07 .PRG .EQU $02E6H ;Parallel Output Handler
00011 BB08 .IOB .EQU 3 ;Current CONOUT: Device
00012 BB09 .SPEED .EQU $0C08H ;Speed Control Byte
00013 BB0A .STAIR .EQU 0FAH ;Status Port General
00014 BB0B .KDR .EQU 1 ;Keyboard Data Ready
00015 BB0C .EOF .EQU 01AH ;End-of-File code
00016 BB0D .TRUE .EQU 0FFH
00017 BB0E ; * Jump Table *
00018 BB0F C3 .EQU JMP CINIT
00019 BB01 C5 .EQU JMP CONST
0001A BB02 C6 .EQU JMP CONIN
0001B BB03 C7 .EQU JMP CONOUT
0001C BB04 C9 .EQU JMP LIST
0001D BB05 C5 .EQU JMP PUNCH
0001E BB06 C2 .EQU JMP READER
0001F BB07 C3 .EQU JMP PRSTAT
00020 BB08 C5 .EQU JMP CONCTRL
00021 BB09 B8B
00022 BB0A 188B

BB18 AF
BB1C 32 0300
BB1F C0
BB20
BB21
BB24 20BB
BB29 DB FA
BB2A 2F
BB2B 06 01
BB2C C8
BB2D 0E 0F
BB2E C9
BB29
BB2A
BB39 298B
BB3B CD 2E00
BB3C CA 298B
BB3F FE 80
BB41 C2 ****
BB43 03
BB47 36BB
BB4F FE 08
BB50 CA ****
BB53 FE 8C
BB55 CA 0600
BB58 E6 7F
BB5B FE 7F
BB5E 0E 7F
BB60 C2 32 ****
BB62 09
BB63 09
BB67 498B
BB68 C5
BB69 06 0B
BB6A C0 0400
BB6C C1
BB6D 53BB
BB6F 00
BB70 00
BB71 00
BB72 00
BB73 C5
BB74 3A 53BB
BB75 87
BB7B C2 ****
BB7C 3A 54BB
BB7D BC 00
BB7E AF
BB7F 54BB
BB80 C3 ****
BB81 C5
BB82 C5
BB83 79
BB84 FE 5C
BB85 C2 ****
BB86 B8B
BB87 06 5F

CINIT: XRA A
STA 10BYTE
RET

; * Determine if console input ready *

CONST: IN STAPT ;Read port
CAMA ANI KDR
RZ MVIP A, TRUE
RET
; * Console input *

CONOUT: CALL KSTAT
JZ CONIN
JMP CPI 80H ;MODE SELECT ???
MVIP A, 03H
RET

NEXT: CPI 8BH ;CLEAR ???
JZ CLEAR
JMP CPI 8CH ;LOAD ???
MVIP A
RET

CLEAR: PUSH B
MVIP B, 0BH
POP B
MVIP A, 0DH
RET

; * Console output *

CONOUT: PUSH B
LDA DELFLAG .BYTE 0
BKLASH .BYTE 0
LASTCHAR .BYTE 0

DELFLAG .BYTE 0

; * Routine to clear screen *

CLEAR: B
MVIP B, 0BH
LDS SCREEN
MVIP A, 0DH
RET

; * Console output *

CONOUT: PUSH B
LDA DELFLAG ;Routine to translate a
ORA A
JNZ DEL ;DEL (7FH) from the
LDA BKLASH ;destructive backspace on
ORA A
JZ RETCH ;This routine works under
LDA BKLASH ;two circumstances --
STA ;First, when the DEL
JMP ENDOUT ;results in an <echo>

DELETE1: MOV A,C ;char) as produced by CP/M
JMP筚C ;for second, when the DEL
STA BKLASH <echo char><backslash>
JMP ENDOUT ;jas produced by Microsoft

DELETE2: MVIP B,5FH ;BASIC.
CONCTRL: CPI 18H ;CTRL-X ?
  JZ XCHANGE
CPI 1SH ;CTRL-S ?
  JZ PAUSE ;The SPACE BAR may also
CPI 20H ;be used to temporarily
  JNZ SCROLL ;halt console output.
PAUSE: CALL CONIN ;Any key resumes the
  RET .console output.
SCROLL: CPI 3AH ;This routine may be used
  JZ XCHANGE
  JZ PAUSE ;to regulate the rate
  CPI 30H ;of console output. During
  RC output, enter any number
  ANI 0FH ;between 0 (fastest) and
  RAL ;9 (slowest) to change
  RAL ;rate. Default=0.
  NOP ;Add another RAL for more
  STA SPEED ;delay.
  RET
; Routine to change console output device
XCHANGE: LDA 10B0YE ;The CONOUT: device may
  ORA A ;be changed during output
  INVERT ;by typing a CTRL-X.
  XRA A ;Thus, a partial listing
  JMP ENDXCH ;may be directed to the
INVERT: CMA ;printer.
ENDXCH: STA 10B0YE ;XRA A
  RET
; * Printer on-line *
PRSTAT: XRA A
  RET
; * Printer driver *
LIST: PUSH B
  MOV B,C ;SOLOS expects char in B
  CALL SDROT ;For a parallel printer,
  CALL POP B ;change to PROUT (CIEXH).
  RET ;Returns with char in A
; * Reserved *
READ: MOV A,C
  CALL MO
  RET
; * Reserved *
PUNCH: RET
; END

Dr. J. E. Byram
27 Bencliff Circle
Newton, MA 01746

RANMODIZE FOR PT BASIC

Have you ever noticed that every time you load a BASIC
same, the random number generator starts in the same place? Have you
ever noticed that some BASIC programs 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 use 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 our 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 UNIT statement for the keyboard port. So
how do we know when the player has started typing his/her
name? We ask the port directly:
  68 REM THE FOLLOWING ROUTINE RANDOMIZES
  78 PRINT "WHAT IS YOUR NAME?" ;
  80 LET C=INP(252)
  90 LET L=INP(252)
  100 LET RANDY=D
  110 IF C=L THEN 90
  120 PRINT CHR(13)
  150 INPUT "..."
  140 LET N=(CHR(1)+N)
Try it. You’ll like it!
file,piolot,mem

Letter to The Editor
Nov. 26, 1984

Bob Sten wasted a way to save and load PTC PILO\T programs in CP/M
(PODIUS NEWS 2(3):10). Lewis Moseley, Jr. suggested using the
incidental PILO\T editor for writing letters (PODIUS NEWS 2(3):2).
I have been using PILO\T together with a custom printer driver
to write voluminous psychiatric reports and have made a quick and
dirty interface with CP/M.

If PILO\T is simply saved on diskette any associated programs will
be lost when it is initialized from location 100K. A simple
construction eliminates this quick forward. KEE\T PILO\T from cassette
to load and initialize it. Set memory size to the desired. I choose to
SET MAV 1800. Type the command 'INFO' and write down the
memory location given at the top line. This will be needed for
writing to disk. Now write PILO\T back out to cassette using the
command 'CUSTOM'. This maneuver modifies PILO\T so that it will
execute the PILO\T program upon initialization rather than
destroying it. Read the castrated version of PILO\T back into
memory. The customized PILO\T will execute rather than erase memory
when the SEARCH command is given. To remove memory, key control-
1, then MODE SELECT, then RETURN. Memory size cannot readily be
changed.

Refer to the PILO\T memory location that we wrote down. Take the
high-order hexadecimal digit, multiply it by 4 and add the second
digit. My memory location was 4970, so I attained 4 x 10 + 9 = 73.
This is the number of 256-byte pages of information that must be
saved. Give CP/M one command 'SAVE 73 PILO\T.COM for whatever
your number of pages was and in 3 seconds you will have a file
named PILO\T.COM which you can load and execute by typing PILO\T.

To save a program, construct a PILO\T program in the ordinary
fashion, then key MODE SELECT to return to the PILO\T command mode.
Enter a backslash followed by 'CALL0' to load a warm boot of CP/M.
You can now save your program via the CP/M command 'SAVE 73 PILO\T'.
Entering 'MODE' will then cause CP/M to run your program. Obviously, each time a program is saved, the entire PILO\T
interpreter and editor is saved along with it. This is an
inefficient use of disk space, but at least it works. The CAPE
save features of PILO\T remain operational.

It is useful to have the first line of a PILO\T 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.

Cleaning PILO\T programs is easy under CP/M. Create a SUMMIT file
listing the names of PILO\T programs (without the .COM attribute)
in the order they are to be executed and use the CP/M 'SUMMIT'
command. Each PILO\T program to be chained should end with the
command 'CALL0', which causes a warm boot of CP/M and the
invocation of the next PILO\T program on the SUMMIT file.

Because a line of PILO\T program which has no color is typed out
when the program is run, PILO\T lends itself to use as a word
processor. Text may simply be typed in. The command 'SET0' anywhere along in the program will divert subsequent lines to a
printer attached to the serial interface, and 'SET0 bottoms restores
output to the video screen.

For fully formatted and right-justified printing I wrote a custom
writer for my Teletype 33 which interprets output from PILO\T one
character at a time. To send the output to this driver I use 'SETO
0X3'. The PILO\T facilities readily print personalized form

letters to be printed. I have certain skeleton reports available
on disk which I read in and then tailor for particular purposes.

Many of PILO\T's features rival or outshine those of the
Electronics Pipeline. Lacking are a black move command, a lincup-
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 leaning 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
caracter. This is done by striking the LOB key. The form-feed
character echoed to the screen may be removed by the DEL key.

If a colon is included in the text, PILO\T 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 PILO\T program being
edited. I hope to create this 'macro' feature shortly.

I would be grateful for any suggestions and for PILO\T source
code (some of which was printed in Mr. Sobeck's) so that I could add
some commands and rework the editor into a super word processor.

Sincerely,
Myron Pullier, M.D.
101 Cedar Lane
Teaneck, New Jersey 07666

TYPE BPILO\T,PRM

* P I L O T

* 29MA860

* PILOT upgrade by Myron Pullier, 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 LOB key.
* In any case, a colon is automatically inserted at the
* beginning of each new line.
* PILO\T must be modified by 'SAVE060 CP/' to use this
* input driven.
* To install PILO\T via CP/M execute
* A:BPILO\T.HEX
* Then, in CP/M
* * must go
* Next, A:PILOT
* Then alter ICFA using SOLOS
MYRON PULIER CONTINUED FROM PAGE 27

<table>
<thead>
<tr>
<th>I/O</th>
<th>REAL</th>
<th>EQU</th>
<th>OSBOM</th>
<th>;CURSOR POSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>ORG</td>
<td>OCBOM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>3532C9</td>
<td>PILF</td>
<td>LIA</td>
<td>F1M2R</td>
</tr>
<tr>
<td>003</td>
<td>FE20</td>
<td>CPI</td>
<td>ER</td>
<td>IS IT A ER?</td>
</tr>
<tr>
<td>005</td>
<td>C409C9</td>
<td>JZ</td>
<td>INSERT</td>
<td>YES, RETURN A COLUMN</td>
</tr>
<tr>
<td>008</td>
<td>FE20</td>
<td>CPI</td>
<td>TRIB</td>
<td>IS IT A TRIB?</td>
</tr>
<tr>
<td>00A</td>
<td>C409C9</td>
<td>JZ</td>
<td>INSERT</td>
<td>YES, RETURN A COLUMN</td>
</tr>
<tr>
<td>010</td>
<td>C08C9</td>
<td>EQU</td>
<td>BREAK</td>
<td>ELSE INPUT KEYPAD DATA</td>
</tr>
<tr>
<td>013</td>
<td>325532</td>
<td>STA</td>
<td>FORMER</td>
<td>SAVE IT FOR NEXT TIME</td>
</tr>
<tr>
<td>016</td>
<td>C9</td>
<td>RET</td>
<td>;AND RETURN IT IN A</td>
<td></td>
</tr>
<tr>
<td>019</td>
<td>3D</td>
<td>INSERT</td>
<td>DCR</td>
<td>A</td>
</tr>
<tr>
<td>01C</td>
<td>36A</td>
<td>MOV</td>
<td>A,COLUM</td>
<td>DELT A COLUMN</td>
</tr>
<tr>
<td>01F</td>
<td>C4C3C9</td>
<td>STA</td>
<td>FORMER</td>
<td>SAVE IT FOR NEXT TIME</td>
</tr>
<tr>
<td>022</td>
<td>09</td>
<td>RET</td>
<td>;AND RETURN IT IN A</td>
<td></td>
</tr>
<tr>
<td>025</td>
<td>B3E0</td>
<td>BREAK</td>
<td>CALL</td>
<td>HSTAT</td>
</tr>
<tr>
<td>028</td>
<td>C1B9C9</td>
<td>JP</td>
<td>BREAK</td>
<td></td>
</tr>
<tr>
<td>02B</td>
<td>FE20</td>
<td>CPI</td>
<td>LOAD</td>
<td>IS IT LOAD?</td>
</tr>
<tr>
<td>02D</td>
<td>C409C9</td>
<td>JZ</td>
<td>FLIP</td>
<td>YES, FLIP TOGGLE</td>
</tr>
<tr>
<td>030</td>
<td>FE20</td>
<td>CPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>033</td>
<td>C0</td>
<td>RND</td>
<td>SPACE</td>
<td>END, RETURN SPACE</td>
</tr>
<tr>
<td>036</td>
<td>C454C9</td>
<td>STA</td>
<td>TOGGLE</td>
<td>TOGGLE SET?</td>
</tr>
<tr>
<td>039</td>
<td>F2C9</td>
<td>CPI</td>
<td>LOAD</td>
<td></td>
</tr>
<tr>
<td>03C</td>
<td>C2B9C9</td>
<td>JZ</td>
<td>SPACE</td>
<td>END, RETURN SPACE</td>
</tr>
<tr>
<td>03F</td>
<td>3D</td>
<td>DCF</td>
<td>A</td>
<td>YES, RESET Z FOR RETURN</td>
</tr>
<tr>
<td>042</td>
<td>36D</td>
<td>MOV</td>
<td>A,</td>
<td>;GET CH FOR RETURN</td>
</tr>
<tr>
<td>045</td>
<td>C9</td>
<td>RET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>048</td>
<td>360</td>
<td>MOV</td>
<td>A,</td>
<td>;GET SPACE FOR RETURN</td>
</tr>
<tr>
<td>04B</td>
<td>C9</td>
<td>RET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04E</td>
<td>C454C9</td>
<td>FLIP</td>
<td>LIA</td>
<td>TOGGLE</td>
</tr>
<tr>
<td>051</td>
<td>FE20</td>
<td>CPI</td>
<td>LOAD</td>
<td></td>
</tr>
<tr>
<td>054</td>
<td>C409C9</td>
<td>JZ</td>
<td>RESET</td>
<td>YES, RESET TOGGLE</td>
</tr>
<tr>
<td>057</td>
<td>30</td>
<td>MOV</td>
<td>A</td>
<td>END, GET LOAD CHAR</td>
</tr>
<tr>
<td>05A</td>
<td>3524C9</td>
<td>STA</td>
<td>TOGGLE</td>
<td>TOGGLE</td>
</tr>
<tr>
<td>05D</td>
<td>C9</td>
<td>RET</td>
<td>;RETURN WITH LOAD CHAR</td>
<td></td>
</tr>
<tr>
<td>060</td>
<td>3C</td>
<td>RESY</td>
<td>INK</td>
<td>;CREATE A SET-CODE</td>
</tr>
<tr>
<td>063</td>
<td>3254C9</td>
<td>STA</td>
<td>TOGGLE</td>
<td>STORE TOGGLE WITH Z</td>
</tr>
<tr>
<td>066</td>
<td>3D</td>
<td>HCF</td>
<td>A</td>
<td>;INSTORE LOAD CHAR, K81, Z</td>
</tr>
<tr>
<td>069</td>
<td>52C9</td>
<td>RET</td>
<td>;RETURN WITH LOAD CHAR</td>
<td></td>
</tr>
<tr>
<td>06C</td>
<td>00</td>
<td>FORMER</td>
<td>DB</td>
<td>D</td>
</tr>
<tr>
<td>06F</td>
<td>0C</td>
<td>TOGGLE</td>
<td>DB</td>
<td>LOAD</td>
</tr>
<tr>
<td>072</td>
<td>55</td>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**BOSTON AREA ANYONE?**

Edwin Meyer wants to know if any Boston area members would like to form a local group for S10 owners. If you are interested, please contact Edwin at 339 Newbury Street, Boston, MA 02115.

---

**ON HELIOS IN A Z80 SYSTEM**

NOV 26, 1979

Dear Stan,

I recently received 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 dealer's magazine 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.

Incised you will find my check and an order for your N series disks. Also included 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 PT DOS with the Helios (I bought one of the first kits at $1800.00) for a few years. I don't have a S10 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 Z80/2AP Disassembler that appeared in the May 79 Dr. Dobbs' 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 written in Z80 assembly language but I don't know how useful 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 PT DOS. It is a go-between program that looks like CP/M to running programs but passes control to PT DOS with properly formatted information. I'm doing this because there is a vast library of programs available from Lifeboat (enough to keep me busy for the rest of my life) but I don't like CP/M. Let's face it; PT DOS 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 their poor quality. It's a relief to have PT DOS.

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 thoroughly 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
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 has been some noise that the drives shipped to customers were out of alignment on arrival. 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 inhabit and repair a number of HELIOS systems, and I have repaired and aligned a number of systems for others. I do technical writing and have experience in operating an 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 Mother 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 the solid-state board, and I know of two or three other component failures. I have put in a device to replace this lamp,

If you need help in getting a system aligned, the HELIOS should have at least some alignment. There is voltage, timing, and phase, but these are not related to the aligned positioner-rod. The device mentioned above can be responsible for some of the misalignments, but certainly not all of them. There are voltage, timing, and phase differences that can be related to the aligned positioner-rod.

I do not want to over simplify the situation, but I think many of the SOL-HELIOS hardware failures are repetitions of a few failures. I have had a few component failures that I think are not related to the aligned positioner-rod.

The biggest problem in the documentation, however, I have found is misalignment of the word-processing system. The J. Reutel is quoted 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 for satisfactory alignment. I remember a lot of time in the beginning trying to determine just how good a wave had to be to prevent computer malfunction.

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 this point, 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 PerSci.

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. Many 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 this project further. On the other hand, if there is a big demand, I can give the project a high priority and turn it out in a few weeks.

Now a word about new systems: I am putting a computer on my SOL-HELIOS System. In addition to output from such as Chalazon and Micro-Soft Basic, I want to run WordStar and take advantage of WordStar's proportional spacing. CP/M is running fine. The printer is a printer driver so that I can send output to the printer. Any printer that will run is a good start; proportional spacing may not be the driver to start with.

I have the IBM R-Type II Printer, Model 1355W. This is the printer. The printer uses the 96 character metal print-wheels; it cannot use plastic print-wheels. It has capabilities of proportional spacing and can use metal proportional print-wheels -- Cubic $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 II printer wheel running on WordMizer.

I am willing to pay for help with a driver which will give me true proportional spacing of WordStar or, perhaps, another word-processing program which has proportional spacing. What I would really like is a revision of WordMizer which has true proportional spacing. I am writing this letter with WordMizer and I like it. The trouble is that I need true proportional

November 9, 1980
...ON UPGRADE FOR HELIOS

September 15, 1980

Dear Stan,

Enclosed you will find a Helios disk with my LOADM program and a 75 character documentation file 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 areas. My principal activities include system engineering, assembly language programming, PDOS 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 EDBASIS 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 PRA C T I C A L 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 interchange-able with other CP/M systems. 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 increased 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 PDOS 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

....ON CONVERTING PDOS 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 PDOS FILE OF UP TO A MAXIMUM UP 8K IN LENGTH TO A CP/M TEXT FILE. THIS IS DONE WITH THE BASIC LANGUAGE PROGRAM CP/M-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 PRO.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 1
4. ZIP (I USUALLY USE ZIP.0)
5. READ FILE,5000 (RECORD THE HEX LENGTH OF THE FILE)
6. LOAD BASIC AND CP/M-TXT
7. RUN (DO NOT SCRATCH OR ALL WILL BE LOST)
8. BYE
9. IMAGE FILE,1400,7000,>LGTH
10. ZIP
11. READ FILE,100,>LGTH (THIS PUTS THE FILE AT CP/M LOCATION)
12. REMOVE PDOS DISK, RESET, INSERT CP/M DISK AND BOOT
13. SAVE N FILE.TXT (SEE CP/M MANUAL TO DETERMINE n)

YOU WILL 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 DECODE 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.I. 5/28/79

10 REM****CP/M-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 PRINT "BEGIN. IT SHOULD BE 20480"
80 INPUT "DECIMAL ADDRESS OF END OF THIS FILE : " ; Y
90 LET L=X
100 FOR I=0 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=2 THEN PRINT "FINISHED"; END
170 NEXT I
180 REM***ADD A LINEFEED FOR CP/M
190 POKE L+8192,10
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***CP/M-TXT

Glo Zeratsky
813 Inlet Road
Green Lake, WI
54941

evenings
714-630-5672
Dear Stan:

First, please accept our thanks for taking on the editing job for Proteus again. You have redoubled the efforts of all members who were already more than one could expect from any ten women! 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 solve this out.

I have a problem which may become more common as more of those with VBM-I's move up to newer CPUs. It is solvable and that someone can help! I have purchased the Godbout Dual Processor board and have discovered that the VBM-I does not work at all with the 8085 at 5MHz, and occasionally malfunctions at 3MHz (usually with fast machine language). I called Godbout, and apparently the problem is that the VBM-I does not meet the IEEE 5-100 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 2-80. My VBM-I works fine with a 4MHz 2-80 or 2MHz 8080. I have not yet been able to try the 8088 (no software yet) but am looking forward to FASC/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! Do love the VBM-I and would hate to have to buy a terminal!

As others who started with CUTER/Inesi systems have undoubtedly done, I have expanded my system to include the Discus 2D and CP/M 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 ROMs and a full set of software when I moved the disc controller to IBM.

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. I have the PTC source in CP/M compatible form, and can use the Discus auto feature to boot CP/M, load CUTER, and then use CUTER for IO and control of software. This all is necessary because the IBM board with CUTER in ROM decodes all addresses for the ROM ALSB and so you can't use the CP/M 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 address space (not changed by the VDMEM routine) so watch out if you move it from O'Connor.

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 our 8085/VDM problem.

Sincerely,

William T. Hole
260 Collinswood
San Francisco, California
94114

---

Dear Mr. Van 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 CPU 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 braided coax cable for my grounding straps, in addition, all peripherals such as printers are also grounded to the CPU via a good ground strap. You would be surprised how many subtle problems disappear when adequate grounding is used. I then have the CPU grounded to power system through a braided cable also, since the three wire 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 PCI boards are a bit tricky in this regard. I am currently using a 48K dynamic memory board made by MEASUREMENT SYSTEMS AND CONTROLS, INC. This board has given no problems at all. I have two of them, one running in my CPU, and the other running in a Z80 machine I designed, that thinks it is a 6502. 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 512K. The 48K version is sold by MINI MICRO MARK of RWAMUSE, NY for $59 assembled, tested, and with a 1 year guarantee. (The 48K board can be expanded to 512K by adding 114K memory chips, which are about the least expensive 1K chips around). A nice feature of this board is that you can select 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 MICROHATION controller that utilizes the READY lines for synchronizing data transfers. It also worked with my VISTA controller for 5 1/4 diskettes.

I wish you the best of luck in getting all your problems worked out, and hope that when you finally get it all running, that you find this software useful.

Sincerely yours,

Th. Thomas McGhee
Fr. Thomas McGhee S.D.B.

Fr. Thomas McGhee 292 Union Ave., Paterson, N.J. 07502

PG. Try using coax braid for your grounds; this may fix your memory problem as well! The bigger the ground strap the better.
Last year, I had volunteered to do a column on hardware compatibility 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 its 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 Mr. McLean in last year's Kilocaud article.

They claim the system comes with the appropriate software and hardware to be able to literally plug it in and be up and running.

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 FT went out of business, Vista bought up as many parts as they could from FT. 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 1K Lynabyte board, which claims to be DMA-compatible, 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 PRM 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 PRM at no extra cost as long as they were told about your requirement at the time of order, and all Lifeboat software 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 tool)

I have talked to a couple of TKS-16 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 you 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 FOR-NEXT statement described by Bob Werner in PROTEUS NEWS, vol. 4, 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 Processors Technology Extended Cassette BASIC Update 731064, page 7. The update has no copyright notice:

Load BASIC into memory using GET.
ENTR the following at DB50: CI CR 40 OB
ENTR the following at JFB1: FE 9D
SAVE BASIC 0 JFB4

Yours sincerely,

[Signature]

Albert S. Woodhull
33 Enfield Road
RFD 2
Amherst, Massachusetts 01002
1 August, 1980
Dear Stan,

I have acquired quite a lot of information from Proteus News, and it’s about time I add my two cents.

My SCL has 5D Sales Expanderom board and I have had no problems with it after the modifications I read in the news letter. The modifications I read in the news letter are great help for someone who doesn’t like the location of a program or just wants to change it. It’s a great help. Everyone in the news letter has done a great job in the best of us with the little problems that occur in programming, 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 there someone out there who can help us understand it all? Has anyone figured out what the display expansion jack (45) 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 card. It allows you, under complete computer control, to generate an infinite 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 AY-3-8910 sound chips. It’s amazing what this board can do. If anyone wants a good sound card 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 code I wrote:

```
START LXX  H, INF  #FETCH A CHAR.
        LAX  D, OUTF  #PRINT A CHAR.
CALL   SQN  #SCL EQU 0000H
CALL   SOL  #SCL EQU 0000H
INF    IN  K5  #SCL STATUS EQU OFH
ANI    IY  #ON DATA AVAILABLE.
JNZ    INF  #LOOP
IN     IN  DATA  #DATA EQU OFCH
ANI    OYF  #STRIP PARITY
GET    MOV  B, A  #READY FOR OUTPUT
CALL   SOUT  #SOUT EQU 0018H
GET    END
```

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

---

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 when 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 reset 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. SYS>(B). 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 "00 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 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 reimage BASIC. Section 5.2.1 of the PTDO 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 3B and 0B in your version, you may be able to find the correct bytes by finding the event CD BC BC 00 (Call SYS->CMEO) in the code and then backing up. The sequence starts at 1E55 in my version, and appears in only one other place, at 0891 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 BACAS 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=4.

Sincerely yours,

Jay Parsons

---

...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 BACAS, SAVE
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 $10. 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 8888 or 280 computer having memory-mapped video? It prefers 16264 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-28 software, but some of it is just too good to lie down to just one machine. (All my software does, however, come initialized for the SOL-28.)

Lewis Moseley sent me a copy of Richard Greenlaw's TAPEDISK routines, including MPUT. 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/CP/M 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,

F. Thomas McGehee

---

NORTHEAST/SOL SOFTWARE FOR SALE

USER AREA FOR DOS: All versions listed include the following features: Run on SOL-28 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 are always accessible! Output devices may be specified via "A" register as follows: 0=VIDEO 1=SERIAL 2=PARALLEL 3=VIDEO+PRINTER. Note that Control/P overrides 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. (Allow 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, SELECTERIC, or any other terminal that does not have overflow problems.

NSTARTSER 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.

NSTART14 is for HEATH MODEL 14 serial printers attached to serial port #1. Provides handshaking via Control/Q and Control/S. Allows operation at any baud rate without overflow.

NSTART15 is for DIABLO 1610/1620 and compatible printers. By compatible I mean any serial printer on port #1 that allows handshaking using the ETX (#3) and ACK (#6) codes. Allows operation at any baud rate without overflow. Assumes the printer has a buffer. (I set for 60 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 $15. Customized versions cost $15. Add $5 if outside continental USA. Price includes instructions, hardcopy listing on sheet paper, and program on inexpensive cassette in SOL/SOL/CUTER format at 1200 baud. Tape is in two parts: USER AREA and PATCH for DOS JMP PAGE. Orders normally shipped within 24 hrs. Send order to: Fr. Thomas McGehee PHONE (201) 595-8800 Don Bosco Tech 202 Union Ave. Paterson, NJ 07502

*** IF DOS ORIGIN NOT 2000, PLEASE GIVE ORIGIN ***
June 6, 1988
Northstar Users Association
111 Highland Ave.
Vesuville, 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 of 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 user 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 (201/595-8880). 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 S.D.B.

PG: 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-23 Users by
Fr. Thomas McGahee
202 Union Ave.
Paterson, NJ 07552

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 8990 (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 5003 (CR). The computer should load the program USER starting at 5000H. (It is OK to load for 2900H, but we are loading into the DOS image we have in memory at 5000H).
9) Type GET JNPS 5000 (CR). The computer should load the program JNPS starting at 5000H. (Again, this is into the DOS image at 5000H).

At this point, the new USER I/O routines have been loaded, and the Northstar JNPS 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 2900 (CR). This will reenter Northstar.
12) Type SP DOS 5000 (CR). This will save the DOS at 5000H onto the diskette.
13) Reset SOL using REPEAT-UPPER-CASE.
14) Boot up Northstar by typing EX 8990 (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/I This allows a new PAGE to be forced at the printer. It can be given ONLY at the keyboard.
BACKSPACE The (CONTROL/I) 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.
LEFT-ARROW  This is changed to BACKSPACE, 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 CONT. 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 function 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:

1) The VIDEO DISPLAY.
2) The SERIAL PRINTER WITH PAGING.
3) The PARALLEL PORT.
4) Both the VIDEO and PRINTER (WITH PAGING).

In addition to accessing output devices by using their pseudopert numbers, CONTROL/P and CONTROL/V allow the current pseudopert to be set from the keyboard. The user may also set the current pseudopert number by stuffing the number into memory at &0B874. In BASIC this may be done via an instruction such as "PRINT $127, &3; DO NOT ATTEMPT TO USE THE SOL'S "SET" COMMAND TO SET THE PSIPORT 1 PRIOR TO Booting UP NORTHSTAR, as CONTROL MAY TERMINATE WITHIN SOLS 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 58. 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 bytes at $2F8F 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/L from the keyboard will immediately reset the page counter and force a FORMFEED operation 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/P to signal a forced NEW PAGE, rather than the FORMFEED character. THE CHARACTER IMMEDIATELY FOLLOWING A CONTROL/P WILL BE USED TO REACTIVATE PRINTING FOLLOWING THE NEW POWERED 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 5800 (CR). This will load a memory image of the DOS at 5800.
2) RESET the SOL using REPEAT/UPPER CASE.
3) Type FN 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 FX 2020 (CR), to re-enter NORTHSTAR.
6) Type SF DOS 5808 (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 NISTARUL NORTHSTAR/SOL/TTY-LIKE PRINTER

NISTARUL 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 299A with a REP instruction. This replacement can be performed on a temporary basis using POKF 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 LDLVL 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 2901. These may be used to extend the CONVERSION TABLE that begins at 29D5. To extend the CONVERSION TABLE, start new entries at 2900, and place REP after last entry. Remember that the most significant bit is stripped only after APTF conversions have been performed, so that the special SOL keys can be detected.
NORTHBASER DOS RELEASE 5.13 has a bug, in that when giving the message "PRESS RETURN TO CONTINUE", it is eliding a CR/LF. It is helpful to replace the "TO CONTINUE" portion with spaces and a terminating CR/LF.

BASIC programs may access the printer via "FILL 51287, 01", and deactivate the printer via "FILL 51287, 80". BOTH the VIDEO and PRINTER may be activated by using "FILL 51287, 81". DO NOT use SOLOS "SET" commands before booting up NORTHSATR, Use CONTROL/P and CONTROL/V to switch the printer on and off, or make sure that the proper value is stuffed into MC87H (51287) via BASIC.

My thanks to Dr. Carl Roth for his suggestions for improving upon my original program and documentation.

ADVERTISEMENTS:

I have custom USER routines for NORTHSATR that include drivers for a wide variety of printers operating under control of a SOL-28. 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.
LOAD may EASILY ADD AN OWN CONVERSIONS.
CONTROL/P can be used to cause PRINTER to REREAD VIDEO.
CONTROL/V can be used to return to VIDEO-ONLY mode.
CONTROL/P AND V ARE AVAILABLE AT ALL TIMES, EVEN DURING LISTINGS.
OUTPUT DEVICE may be selected as device 0-3.
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 with most TTY-type terminals.
PRINTING MAY BE PAUSED.

PRINTERS SUPPORTED

NORTHSATR: Connects via SOL Serial Port 1. Provides delay following a linefeed by sending out a user-defined number of NULLS. Comes set for 10 NULLS. FORMFEED is accomplished with linefeeds. MAY BE USED WITH ANY SERIAL PRINTER.

NORTHSATR: Connects via 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.
SLOWED SERIAL 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 printets that tend to overfill their buffers. The small delay can be "tweaked" by the user to allow serial printers to run at higher baud rates without loss of data. FORMFEED is accomplished with linefeeds.

SERIAL HEATH 114 via SOL Serial Port 1.
SERIAL HEATH 114 INCLUDES HANDSHAKING using Control/Q and Control/S.
SERIAL HEATH 114 operation at ANY baud rate without loss of characters.

SERIAL DIABLO 1520 (or equivalent such as IPSI 1522)
SERIAL DIABLO 1520 connects via SOL Serial Port 1. Allows use at ANY baud rate without loss of data. Uses ETX (03) and ACK (06) 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 shoe-string budget here, and rely upon announcements in club newsletters, and word of mouth to provide advertising.

Currently stocked USER routines sell for $10 in continental USA, $15 elsewhere, including CANADA. Routines written for printers not currently stocked cost $15 in continental USA, $20 elsewhere.

Send check or money order to:
Mr. Thomas McGane
202 Union Ave.
Paterson, New Jersey 07502

Other programs available (These run on ANY 8800/280):
MODIFIER II: utilities for 8800/280 and memory-mapped video. For folks who like to dig around inside programs. Includes an easy to use memory viewer on screen. Live/Enter screens into memory. Find/Dump/Enter hex. Display pages of memory on screen. And more.

SOLOS: an on-line mini screen editor for 8800/280 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 BASE for full details.

BUG IN GERRY FRICKE'S FOOTBALL

You might note that there is a minor bug in Gerry Frick's FOOTBALL game on disk M1. 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 backards some of the time.

Keep up the good work,

Jay Parsons
Don Bosco Technical High School
202 Union Avenue, Paterson, New Jersey 07502
Telephone: (201) 758-8600

December 1979

Dear Mr. Van Tienenfeld:

My modification involves installing an optically-isolated zero-crossing solid-state relay in series with one of the motor drive wires. The relay gets power from the inverter and relay will shut off with a fault. I used the two wires I soldered to the original relay (actually, it is a switch) over the solid-state relay. There is plenty of room inside the Thinker toys drive cabinet for mounting the relay. Now whenever the head moves the motor will stop. During idle time, the motor is off. It turns on very smoothly and no transistor noise. I have had absolutely no trouble with this, and I have been able to make this modification. The solid-state relay is stock #5125E-01 from Universal Electronics, Inc. Your address is: 4050 South La Cienega, Los Angeles, California 90044. Orders over $100 require a $3.00 handling charge plus postage for first item. (This is an excellent solid-state relay.)

The modification should work on your system with no problems if you change for the "configuration" just let others know about it if you find it useful.

Sincerely yours,

THOMAS A. AUYON

ON SCIENTI’S "PIMS"

I have been attempting to implement Scienti’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 DEVRDR and USB statements to call the OTHER tape subroutines: OPEN, CLOSE, WRMT, READ, WRBLK and RBBLK, but this does not seem to work properly. Can any reader of PROTEUS suggest a solution to this problem? H.B. I am using the G2 Microsoft Extended Basic version 4.7 rather than PBasic as the latter was not available locally when I bought my C64T and GTP boards.

Yours truly,

Alastair S. Preston
150 Twin Terrace
Bnton, Alberta,
CANADA, T6N 1Y4
23 July 1980

---ON A PROGRAM CORRECTION, ETC.

November 29, 1979

Dear Stan,

Please print the following patch to program P on page 19 of Vols. 2 No. 4:

```
CAFO 70 WRITE MOV A,B
CAFO 71 FE 00 CPI OOH
CAFO 72 FA 08 CPI 20H
CAFO 73 FA 0A XRA A
CAFO 74 FA 01 CB JRP WRITE
CAFO 75 6A 61 CB WRITEO LDA FLAG
CAFO 76 3C 00 00 WRITE1 STA FLAG
CAFO 77 32 61 CB WRITE1 STA FLAG
```

NOTE: SCRATCH the old version and ASSM new version

This patch restricts the 8-bit mode recording to a single (CAO) byte between lines of BASIC program text only making the recordings match the Cassette Software Library standard.

Some general comments:

1. Does anyone know of a place I can get a SOLOG personality module which uses 2708 EPROM’s?
2. What I hear about being able to read TRS-80 cassettes.
3. I have always given it up because of the differences in BASIC language syntax (especially the input/output to the diskless, i.e. high and low resolution graphics).
4. Does anyone know if a SOD-20 MICROFASL version of PASCAL is available or in the works?

Keep up the good work with PROTEUS NEWS.

MELVIN M. DALTON
7826 WEST 30TH ST.
PLAYA DEL REY, CA 90291

---ON HORRORS---

September 21, 1980

MENCO SERVICES
4837 Fairview 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 occasions, 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 a kind of “horror” story. Back in May, John Dvorak announced the availability of a North Star Basic version of “Adventure” which he 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 SOD has 20K 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 occasions (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".

Yours truly,

Alastair S. Preston
150 Twin Terrace
Bnton, Alberta,
CANADA, T6N 1Y4
23 July 1980
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 encountered unexpected obstacles. These are on the road to solution, and are explained in the letter.

In the next issue, we will 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 breaks and blanks. 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. Cassette 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 all compatible files are acceptable, too.

BASIC COMPUTER GROUP SOFTWARE UPDATES

If you have the BC.G'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:

Wordwiz 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 1.0.0
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 0000 address space, with the ability to switch back to 0000 just by reinserting your old personality module. And Stephen Maguire, Joe's son and apparent programming wizard of the family, 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 repair 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, TREP-80, SC BASIC, $1075 negotiable.
George Atwood 2119 Sunite St. Berkeley, CA 94709 415-525-3867

COMPUTER 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

Editor’s Column / Stan Sokolow .................................................. 1
Call For Help on Library Disk H-4 .............................................. 1
Time to Renew All Subscriptions .............................................. 1
New Helios Library Disk Available ........................................ 2
Contents of H-6 Diskette .......................................................... 2
Proteus Software Library News ................................................ 3
News Release: Software Music Synthesis System ......................... 3
News Release: Contest: Personal Computing for the Handicapped .... 4
4th Annual National Computer Camp ......................................... 4
News Release: MIDB & Schema Redesign / Micro Data Base Systems, Inc. 4
Software Vendor Directory / Micro-Serve, Inc. ......................... 5
The Answer Man / Joe Maguire
-- to Henderson: Power Blinks in Nepal .................................. 6
-- to Gorman: One Port and Two Printers ................................. 6
-- to Jim Michaelis: Queue Problem ........................................ 6
-- to Roger Doran: Electric Pencil Problem ............................. 7
-- to Earl Dunham: Wordwizard Printer Drivers ....................... 7
Comments ............................................................................. 7
A Report on Northstar Computers' Hard Disk System / Joe Maguire . 8
Moving ECSO to FODO / Bob Stak ............................................. 8
Another Note on PSCAN / Grayson Evans ................................. 8
What Disk Should I Get / Editor ................................................. 9
Equipment Review: Memory Boards and 280 Processors / Grayson Evans . 9
Minor Improvements to the Helios Disk Controller Boards / Grayson Evans . 9
Modification for Using the Cuts Board as a Serial Port / Steve Elsht .......................................................... 10
Interfacing a SOLO Computer to a WIN-M-14 Printer / Bill Supynuk .......................................................... 10
Device Drivers Explained / Grayson Evans ................................ 10
Micro Communication Program for PTDOs / Michael McKelvey .... 11
Unclassified Ads .................................................................... 11
More Unclassified Ads ............................................................. 12
SoluRef - North Star CP/M User Area Routines for the SOL/20 / Jim Byram ... 23
Randomize for PT/ Richard Bjornstad ................................. 27
Letters ................................................................................... 27
-- on PFC Pilot to Disk / Myron Puller, M.D. ......................... 27
-- on Bonner / Area Answer ...................................................... 28
-- on Helios in a 260 System / Grayson Evans ....................... 28
-- on Per Sci Drives, Helios Service Manual, etc. / Hal A. Lindsey .... 29
-- on Upgrade for Helios / Larry McDavitt ............................... 29
-- on Converting PTDOs Files to Lifeboat CP/M / Glb Zeratsky .... 30
-- on VDS-1 Problem with 8085 / William T. Hole ................. 31
-- on System Noise Problem / Thomas McGahee .................... 31
-- on Vista Disk for SOL / Jordan L. Torgerston ...................... 32
-- on For-Next Loop Fix for EC Basic / Albert S. Woodhull ....... 33
-- on McGahee, Expansion Jack J4, and a Sound Generator / D.F. Siebenrock .. 33
-- on Microsop Users Group .................................................... 33
-- on Hytype, DD Basic, Save / Jay Parsons ......................... 33
-- on Greenlaw's Tapedisk for CP/M / Fr. Thomas McGahee . 34
-- on Custom User Area for SOL/Northstar / Fr. Thomas McGahee ... 34
-- on Bug in Gerry Fricke's Football / Jay Parsons .................. 34
-- on A Disk Modification / Fr. Thomas McGahee .................... 35
-- on Sclaro's "FIMS" / Alastair S. Preston ....................... 38
-- on A Program Correction, Etc. / Melvin M. Daitman ......... 38
-- on Horrors / Menco Services ............................................. 38
PFC Source Code News / Editor .............................................. 39
Call for More Articles and Letters / Editor ............................ 39
Basic Computer Group Software Updates ............................... 39
Coming Attractions ............................................................... 39
Membership Lists Available To Members Now ......................... 39
More Unclassified Ads .......................................................... 39
Table of Contents ............................................................... 40
Subscription Renewal Form ................................................... 40

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.

Copyright (C) 1980 by Proteus. All rights reserved. Permission is hereby granted to reproduce any computer programs contained herein, unless otherwise stated on the program, provided that Proteus and the program's author are given credit.

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

FIRST CLASS
US POSTAGE
P A I D
PERM NO. 655
S. SAN FRANCISCO, CA