PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

JANUARY/FEBRUARY 1981 PUBLISHED BIMONTHLY BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA VOLUME 4, NO. 1

EDITOR'S COLUMN

by Stanley M. Sokolow, B.A., D.D.S.

With this issue we begin our 4th full year of operation as the Processor Technology Users Society. The old timers will recall that the group began in August, 1977, while Processor Technology Corporation was still a rising star in the microcomputer heavens. In 1979, when PTC plunged so suddenly from the sky into the abyss, Sol owners were caught by surprise and began to mildly panic. I personally felt that continuation of the users organization was essential and have worked to the best of my ability in precious spare time to keep it going.

It isn't easy, let me tell you. In addition to the praise and help of members, we get our share of snide remarks. People want gold for the price of garbage, and at times it gets me down. (This is one of those moments.) So I want to clarify what Proteus is and is not.

Proteus is not a tax-exempt, non-profit corporation. I personally don't want to spend the time or money on the red tape necessary to do this, and I don't want the hassles of complying with the regulations. Proteus would best be called a benevolent proprietorship. We employ a half-time person, Jane Delno, who is our secretary, production staff, office manager, bookkeeper, etc., etc., all rolled into one. I am the self-proclaimed Executive Director, resident technical expert, historian, editor, publisher, financier, marketing director, director of new product development, etc.. And that's it as far as official staff goes. We have many contributing helpers, such as Chuck Athey who has done a lot of work cleaning up the Processor Tech Source code we are distributing, Lewis Moseley who runs the cassette library, and so on. But the bulk of the work falls on Jane and myself.

As you can see from the degrees after my name above, I am a dentist (specifically an orthodontist) by profession. The B.A. is in math, obtained at a time when computer science degrees were non-existent. I was heavily involved with computers at UCLA as an undergraduate, ran the UCLA computer club for a while, and was accepted into the Stanford Computer Science Ph.D. program, but changed my career plans before entering Stanford. Nevertheless, I have kept up with computers.

Proteus, for me is somewhat of a personal commitment. The little time I have that is not involved in the demands of my orthodontic practice and the needs of my family, I spend on Proteus. But my practice needs have grown since I started the newsletter, and they have been exceptionally heavy for many months now.

I have chosen to do things slowly but properly, rather than fast and sloppily. Those who are waiting anxiously for things from Proteus should understand this. But realize that if I hadn't kept pushing for such things as access to the source code, and so on, it wouldn't have happened at all. I regret that everything takes so long, but I am cautious about what I farm out to people who offer help and to whom it goes. It is difficult to control an organization run by mail. I welcome help from anyone in the San Francisco Bay area, and would really like to find someone in a 25 mile radius of my office who is willing, interested, and capable of becoming a co-worker. If you fit the description, please contact me.

Regarding the PTC source code, we are beginning to ship copies of the Helios disks. The CP/M disks will be ready soon. The cassettes will take somewhat longer because of the number of people they must go through. I promise they are coming as fast as I can push the software through the steps. If everyone were willing to pay several times as much as we are charging, it would be easy to hire someone to do all of the work and get them out fast. But I have chosen this comprpomise between speed and cost.

Half of the delay since we first announced the fact that we obtained agreement to be licensed to distribute the code was due to PTC's negotiations with a major creditor. We were kept in the dark by PTC on legal advice. What we finally got needed some work to verify its validity and repair some problems discovered. Even as late as a week ago, I obtained a newly discovered disk of more recent versions of certain programs. The delay is resulting in improvement, so hang in there.

Well, I could go on, but it's probably too much already, so bye for now. See you at the 6th West Coast Computer Faire, where we can discuss some of these things.

Sincerely,

Stan

Stanley M. Sokolow

See Us at the Computer Faire April 3-5

PROTEUS GOES TO COMPUTER FAIRE

Once again Proteus will be represented at the West Coast Computer Faire, April 3-5, Friday thru Sunday. The Faire management has kindly donated Booth 1448, near the food concession.

We also have a meeting room Sunday morning 10 am to 12 noon, before the Sunday exhibit hall hours. If you have not yet made plans to attend the Faire, you should reconsider. It is going to be the largest ever, and it should be fun to see what new products are coming up.

The Faire will be in Brooks Hall and the Civic Auditorium, San Francisco, California. It will be heavily advertised on a nation-wide scale. Even IBM will be there again. Tickets are $10 for the entire show. Contact Computer Faire, 333 Swett Road, Woodside, CA 94062, telephone (415) 851-7075.

NEW PRODUCT NOTES

While it is not a brand new item, you should give consideration to the Epson MX line of printers if you want a quality printer at an excellent price. We've used the MX-80 for over 6 months now and have only high regard for this printer. It is relatively quiet, 80 characters-per-second, bidirectional, logic-seeking (takes shortest path to printing of next line), adjustable tractor, compressed characters, expanded characters, horizontally emphasized, vertically emphasized, etc., etc. From mail order houses I've seen it advertised for $550 to $600. Now there is a stripped down version, the MX-70, that doesn't do logic seeking (it always prints a full line width) and doesn't have lower case, but it sells for about $450 from discount houses. The serial-interface version will work with the Sol3 driver in AccPac, MailMaster, & WordWizard. The parallel-interface is Centronics compatible.

Digital Research has announced availablity of a record retrieval system intended for use with their PL/I-80 compiler. Called BT-80, this system uses the B-Tree index technique which is superior the the "ISAM" structure commonly used now. It provides a self-reorganizing tree structure for the indexes into the file, concurrent access of up to 4 indexes, etc.. Requires 48K CP/M version 2, MP/M, or CP/NET operating systems, as well as PL/I. $200 for BT-80. PL/I-80 with compiler, relocatable assembler, linker, is $500 on standard 8" CP/M diskettes, OR NORTHSTAR (Double Density), OR MICROPOLIS! (I didn't know they supported other disks. Wow! That's a surprise.) Digital Research, Box 579, Pacific Grove, CA 93950.

Digital Research also announces version 1.3 of PL/I-80. They will update older versions for a $25 service charge. This version includes two new features: PICTURE specification for output formatting, and a detailed supplement explaining the use of PL/I in commercial applications with financial program examples. PL/I-80 was reviewed in Sept. 1, 1980, InfoWorld.

Micro Complex produces a NorthStar-compatible controller board which acts like the NorthStar board, but only better. It will allow double-density recording in the track, 96 tracks per side, and two-sided drives. This will upgrade a NorthStar system to 1.6 Megabytes using the new controller and 2 Tandon TM-100-4 mini-drives. The DOS is fully compatible with NorthStar DOS, so existing programs will run. Micro Complex, 25651 Minos Street, Mission Viejo, CA 92691, telephone (714) 770-2168.

Metron Computerware, Inc., is selling Sol 2708 personality modules which allow you to move your Sol's dedicated address space from C000 to F000; yet by re-inserting your present module, you can go back to the C000 standard. (See next paragraph for significance of this feature.) Requires minor modification to Sol. $129 plus $3 shipping, assembled, including 2 2708 EPROMs programmed with relocated SOLOS. Metron Computerware, 552 West 114 Street, New York, NY 10025.

Micro Complex also has announced a new personality module for Sol which will let you use 60K of contiguous RAM, by relocating the Sol's dedicated address space (video RAM, system RAM, ROM) to F000. We've seen something like this before, such as the one from Metron Computerware, but this one is better. It has space for two 2716 EPROMs which are selected by a switch on the back edge of the module, accessable with the Sol cover closed. The switch not only selects which ROM, it also selects between the C000 and F000 locations. So you can go back and forth between standard Sol and the "stretched" version at the flip of the switch. Customized versions of the Solos are available for NorthStar disk controllers, Helios controllers, both controllers, etc. It only requires the usual simple jumper modification of the Sol board. The best part is the price. $95 including the assembled module, two programmed PROMs, and documentation. Add $3 handling and California residents add 6% sales tax. We have received a module for evaluation, so we know it is a real product, not just a pre-development advertising campaign to test the demand.

There's lots of life left in the ol' Sol yet.

Teletek announces the I2 Intelligent Interface, a bidirectional parallel I/O port board designed to simplify interfacing with intelligent hard disk or tape controllers such as the BASF, Burroughs, Century Data, CDC, Corvus, IMI, Kennedy, Memorex, Micropolis, Pertec, Priam, Shugart Associates, etc.. Also includes EPROM for driver routines, RAM buffers. Single-port version $325 quantity 1, dual-port $350. Teletek, 9767 Business Park Drive, Sacramento, CA 95827, telephone (916) 361-1777.

SofTech Microsystems announces release of UCSD p-System version IV.0 and release of a compatible BASIC compiler. The new version offers consolidation of features found in the UCSD Pascal system off-shoots (Apple, WD MicroEngine, & II.0). Upward compatibility is maintained. Allows compilation of larger programs, automatic swapping of operating system Segments, user control of Segment residency optional, multi-tasking primitives, unlimited number of Units and Segments, code sharing, command I/O redirection, etc. Also available are additional cross-assemblers, for 6809 & Z8.

The BASIC compiler includes long variable names, optional line numbers, Include files, virtual arrays, compatibility with UCSD Pascal & Fortran-77.

SofTech Microsystems, 9494 Black Mountain Rd., San Diego, CA 92126. (714) 578-6105.

Proteus now has listings of the PROMs on the Processor Technology nKRA RAM board (64KRA). Buyers of the Advanced Computer Products special who need to burn their own PROMs can get a photocopy of the listing from Proteus for $5. It's not elegant, but readable.

McGraw-Hill, publisher of Byte magazine and "onComputing," announced plans to change "onComputing" to a monthly called "Popular Computing" and make a major push into the "exploding field of microcomputing." It will aim at the intelligent, but non-technical readers who are expected to be the masses buying the emerging low-cost microcomputers. "onComputing" circulates 70,000. Byte circulates 150,000. Popular Computing will be direct-mail advertised to 1,500,000 prospective subscribers. Getting big.

I wonder how long it will be before every house comes with an RS-232 interface?

SOL SYSTEMS AVAILABLE

Don Cooper of FD Systems, Inc., has SOL systems available for purchase, these include: Helios III systems and SOLs with North Star drives. Contact: Don Cooper
F D Systems, Inc.
P.O. Drawer P
Stafford, Texas 77477

Repairing your Sol

by Joe Maguire

Even if you have never held a soldering iorn or screwdriver in your hands before, you can learn something from this article. I'll give you some tips for trouble shooting a sick Sol that require no use of test equipment. Most failures can be found with nothing other than a Volt/Ohm meter. There may even be a few tibits for the experienced technician.

When the demise of PTC was announced in these pages some time ago, the groans of concerned owners could be heard around the world. The most concern centered around required repairs. Where to go for getting those inevitable breakdowns fixed? It may be that you need go no farther than the nearest table top.

I have been building and repairing Sols since PTC first started producing them. I think I can say that I have seen just about every problem that can happen under that pretty blue cover. With just two exceptions, every sick Sol I have encountered could be fixed in the average hobbiest's workshop. (Of those two, one had been hit by lightning and the other was my first experience with a malfunctioning 16KRA memory board - ugh! More about that later)

First Step: Preventative Maintenance

You know the old saying about an ounce of prevention. Well, change that to: a few minutes time may be worth many, many dollars! The big bugaboo of the Sol is heat. More failures can be traced to this than any other cause. The cooling fan in the Sol is just marginal at best so regular cleaning is required.

CLEANING

One of the best cleaning tools is a paint brush. Get one about one inch wide. Get a second brush with stiffer bristles for tough spots. A good choice is a brush used for meat basting. Take the covers off the Sol, remove all the boards from the backplane, and carry the chassis to a spot outdoors. (you don't want all that dust flying around inside) Now, stand it up on end and liberally "paint" away all the dust and dirt that has accumulated around the ICs and other parts. Dust can act like an insulating blanket around an IC causing it to overheat. For the most thorough job remove the keyboard. (But remember where to reconnect the cable!) If you have a vacuum cleaner with a small tip by all means use it. The same for an air hose but don't use too much pressure. Cleaning the keyboard is a little tricky. The fit between the key assembly and the circuit board is not air tight and rough cleaning can force dirt into the assembly causing keying problems. I use only a vacuum device for cleaning the keyboard. To remove grime and stains between the keys, use a "Q tip" with some window cleaner.

The stiff brush can be used to clean the fan blades, the power supply compartment and wherever else it's needed. Don't forget the memory and other boards you took out.

Tough jobs: Large computer installations have rules about drinks and smoking around equipment. They learned the hard way what smoke particles and liquids can do. That should be the rule in your computer room too - particularly if you are using disk drives. Smoke particles stick to the ICs and circuit boards and dust and dirt sticks to the smoke residue like glue! Getting this mess back to the original pristine condition requires drastic action. The method I use is going to shock you but it works. I dismantle the Sol and put all the parts in the dishwasher! About ten minutes on the short cycle and all is new again. Remember, all ICs are watertight and so are all the other parts except the power transformer. I don't put in any painted parts, the fan, the keyboard, the transformer or the wood sides but the rest is fair game! Let the parts thoroughly air dry after the wash. Do not use the heat dry cycle as this can warp the circuit board. If the metal chassis parts are badly tarnished, I use some Ajax or Comet cleaner with a kitchen scouring pad. To keep these parts from tarnishing again, I wipe on some floor wax after I assemble everything. Don't wax until everything is assembled otherwise some poor electrical contacts might result.

TIGHTENING

Heat causes strange things to happen in the Sol. I have found, over the years, that ICs can just pop out of their sockets due to flexing of the circuit board caused by temperature cycles. Each time the Sol is turned on, used for awhile and turned off, it undergoes one of these cycles. If you use your Sol as often as I do mine, it doesn't take long for an IC to come loose. About once a month I take off the covers and push all the ICs back in their sockets. I do this for all the boards installed in the backplane too as well as the seating of the boards in their S-100 connectors. It's amazing how many ills this simple procedure has cured. I'd call it the number one trouble shooting technique. Don't forget to check connectors and cables while you're at it. Bad cable connections are notorious; particularly ones subject to movement such as the video or printer cables.

Second Step, Isolate

OK, in spite of all the preventative maintenance, once in awhile the Sol just quits. What do we do then? Let's take care of the obvious things first. Has the AC power cord become unplugged? Is it a blown fuse? These problems can be spotted by the fact that the Sol's fan isn't running or that there's no raster on the TV monitor. Fuses usually don't blow without reason and if you discover this you most likely have a power supply problem. But my experience has shown that about 99% of the troubles are due to IC failures. The job is to find them.

Certain types of memory failures, particularly static, can result in the memory board seizing the address or data bus and rendering the computer completely dead. Removing the offending board from the backplane restores basic operation. Therefore, the first step in isolation is to remove all boards except the personality module. (power off, of course) Now what do you have? Can you get the SOLOS prompt on RESET? Do the SOLOS commands such as DUMP work OK? If so, the trouble may be with a board you took out and not the Sol itself. I'll get to boards later but let's say the worst has happened and our trusty Sol is really sick.

The isolation process should proceed in a systematic manner. The most logical place to start is with the power supply. A volt meter is required for this step and things will be much easier if you have the proper schematic diagrams. Measure the voltages at the main power connector (J10) on the Sol PC board. Are they within 10% of normal? If so, move on. If not, you most likely have found the problem, the power supply. One check of the power supply remains, however. It is possible for a short circuit on the main PC to cause the power supply to appear dead. To isolate this, turn off the computer and disconnect the power connector from the PC board at J10. Now turn the power back on and check the voltages at the connector coming from the power supply. If a required voltage is missing, you have isolated the problem. If the voltages are now back to normal, we go on to the next step.

In the next issue: Tackling the Sol PC board.

A Wild Card option for the Memory Search command of North Star's Monitor 5.2

Contributed by Joe Maguire

As promised, here is the wild card search routine for patching into North Star's Monitor program. This routine will work with any release after 5.Ø but is shown assembled for the standard origin of release 5.2. As I indicated in my previous article on adding a Hex Math routine, this one will fit into some of the extra file space reserved on the disk for NEWMON. The listing is self explanatory but perhaps a few examples will help.

```
SM CØØØ-C7FF C3,*,*,C9         Finds C3 and C9 separated by 3 bytes.
SM 2DØØ,FFF "t","h",*,"s"      Finds "this", "thus", "those" etc.
SM 1ØØ-49151T CD,"*",175T      Searches for CD,2A,AF only.
SM Ø-FFFF *,*,*                Will print out 65535 addresses.
```

```
                     ;
                     ; Wild Card option for Memory Search
                     ; routine of North Star Monitor 5.2
                     ;
                     ; Syntax:  SM addr Byte,*,Byte,Byte,*,...etc.
                     ;          The asterisk character will match
                     ;          any byte value.
                     ;
                     ; Example: To find all JMPs to the EØØH addr
                     ;          area, type:  SM addr C3,*,ØE
                     ;
                     ; Note:    ,*, and ,"*", are not equivalent.
                     ;          The first is the wild card.  The
                     ;          second will substitute the Hex value
                     ;          2A in the byte list.
                     ;
ØEØØ                 ENTRY:  ORG  ØEØØH  ;For Monitor at ØEØØH as supplied
                     ;                    on factory master disk.  You can
                     ;                    change this as required.
                     ;
1Ø34                 SMCOM:  ORG      ENTRY+234H  ;Change entry in command
1Ø34 8Ø16                    DW       ZBUF        ;table to jump to this
                     ;                            ;routine
                     ;
11BC =               SMEM:   EQU      ENTRY+3BCH  ;Original SM address
                     ;
11CE                 PACH2:  ORG      ENTRY+3CEH  ;Jump out to check for
11CE 8F16                    DW       MATCH       ;wild card.
                     ;
11D6 =               NMACH:  EQU      ENTRY+3D6H  ;Return here if no match
11DØ =               YMACH:  EQU      ENTRY+3DØH  ;Return here if a match
                     ;
138F                 PACH3:  ORG      ENTRY+58FH  ;Jump out to store wild
138F 9B16                    DW       PUTWC       ;card in buffer.
                     ;
1393 =               BFILL:  EQU      ENTRY+593H  ;Original buffer fill
13A1 =               QUOTE:  EQU      ENTRY+5A1H  ;Check for quote mark
                     ;
168Ø                 ZBUF:   ORG      ENTRY+88ØH  ;Start of Wild Card routine
                     ;
168Ø 21AF16                  LXI      H,WCBUF ;Zero the Wild Card buffer
1683 Ø61Ø                    MVI      B,16    ;Buffer length
1685 36ØØ            ZB1:    MVI      M,Ø     ;Move in the zero
1687 23                      INX      H       ;Bump pointer
1688 Ø5                      DCR      B       ;Decrement counter
1689 C28516                  JNZ      ZB1     ;Keep looping if not done
168C C3BC11                  JMP      SMEM    ;Return to SM routine
                     ;
168F D5              MATCH:  PUSH     D       ;Save original pointer
169Ø 14                      INR      D       ;Bump it up 256 bytes
1691 1A                      LDAX     D       ;Look in WC buffer
1692 FE2A                    CPI      '*'     ;See if it's a WC
1694 D1                      POP      D       ;Get back pointer
1695 C2D611                  JNZ      NMACH   ;No WC, return to routine
1698 C3DØ11                  JMP      YMACH   ;Yes.  Bypass routine
                     ;
169B FE2A            PUTWC:  CPI      '*'     ;See if WC in byte list
169D C2A113                  JNZ      QUOTE   ;No WC, return to CPI '"'
16AØ E5                      PUSH     H       ;This is part of
16A1 D5                      PUSH     D       ;original code
16A2 E5                      PUSH     H       ;Save original pointer
16A3 24                      INR      H       ;Bump it up 256 bytes
16A4 362A                    MVI      M,'*'   ;Move in WC flag
16A6 E1                      POP      H       ;Get back pointer
16A7 C39313                  JMP      BFILL   ;Continue orig routine
                     ;
                     ; Note:  The Wild Card buffer lies 256 bytes above
                     ;        the regular byte list buffer in order
                     ;        that a simple INR can be used to adjust
                     ;        the required pointers.
                     ;
16AF                 WCBUF:  ORG      ENTRY+8AFH
                     ;
16AF                         DS       16      ;Length is 16 bytes
                     ;
16BF                         END
```

Moving SOLOS to FØØØ Hex

by Joe Maguire

Having turned to CP/M out of desperation to find a macro assembler, the next problem I ran into was not enough memory in the Sol. Not that I didn't have it available, I did. But the SOLOS operating system, at CØØØ hex, sat smack in the middle (or so it seemed) rendering the top 16K unusable. CP/M, unlike PTDOS, requires a contiguous memory block. I decided something had to give. I am not about to be bested by a box full of ICs (even if it is my beloved Sol) so with apologies to Lee Felsenstein, (I considered his modification kit too expensive) I dug into the circuit diagrams to see how I could relocate SOLOS, its RAM and the video RAM to a higher address. The total memory required for SOLOS and the VDM is 4K. That made the highest permissible address FØØØ hex.

The hardware modification turned out to be absurdly simple - one jumper! The software relocation may be a stumbling block for some but it is not out of reach of the well equipped hobbiest or as a computer club project. What is needed are a pair of 27Ø8 EPROMs (or single 2716) with SOLOS reassembled to address FØØØ hex and burned into them. Then they must be installed in the personality module in place of the original 9216 masked ROM. As luck would have it, there are a few unused contacts on the personality module connector so, with just one additional jumper, the switch from SOLOS at CØØØ to FØØØ and back again can be accomplished entirely by just exchanging modules. Spare module boards (for 27Ø8s) have been advertized from time to time in various publications including the Proteus Newsletter and Kilobaud/Microcomputing.

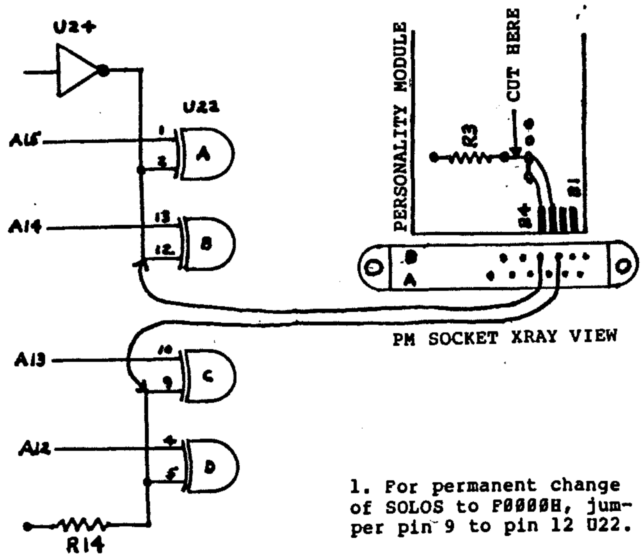
I should emphasize that the relocation moves everything up to FØØØ. The 4K block from CØØØ to CFFF is free for other use. This can add up to 12K additional memory for use by CP/M. (CØØØ-EFFF) Some readers are no doubt wondering about software compatibility but with the quick change capability, this is not really a problem.

The source code for SOLOS is available from the CP/M or Proteus libraries. You can also copy it from the Sol manual if you don't mind all that typing but the original source code contains several absolute address references which can cause problems unless they are changed. See below for corrections.

```
  Original source code                 Modified code

CØDF FE DØ      CPI ØDØH      FE ØØ       CPI Ø        End of Screen
CØE1 DA DB CØ   JC  ERAS1     C2 DB FØ    JNZ ERAS1    Keep blanking

VDMEM  EQU  ØCCØØH            VDMEM  EQU  START+ØCØØH   Screen Adr
```



PM SOCKET XRAY VIEW

1. For permanent change of SOLOS to FØØØØH, jumper pin 9 to pin 12 U22.

2. For change by plug in of PM; on solder side of Sol PC, run jumpers as shown from pins 9 and 12 of U22 to pins B3 and B4 on PM socket.

3. For FØØØH PM; on component side of PM cut trace as shown next to R3.

4. For CØØØH PM; on component side of PM cut trace as shown next to R3 and on solder side of PM cut connection between B3 and B4.

SOLOS modification to FØØØH

UNCLASSIFIED ADS

WANTED: 4 drive Helios system and SOL printer.
Allan Olson, 11321 Ridge Rd., Nevada City, CA 95959
916-265-6675 - evenings, 916-444-3780 - days.

WANTED: 16K or 64I board and Helios II dual disk drive.
W.W.K. Zung, M.D., Duke University Medical Center, Veterans Hospital, Durham, North Carolina, 27705. 919-286-0411

WANTED: A WordWizard compatible printer driver for the SOL 2 metal wheel printer that does true proportional spacing.
Michael A. McKelvey, 330 So. State St., Ann Arbor, MI 48108

WANTED: A driver for Centronics 603 printer to interface with Helios and AccPac/WordWizard software.
Nathaniel Pulsifer, Ipswich, Mass. 01938, 617-356-3530.

FOR SALE: IBM Selectric II (Micro Computer Devices, Seclecterm, System 9710) interfaced for the SOL. I need a faster printer. This selectric has all the options, tractor or friction feed, self-correction, dual pitch and software for a North Star drive. Asking $1,400.00.
Joe Lancaster, 1931 Cedar Ridge Dr. #18, Stockton, CA 95207

WANTED: SOL/HELIOS - Complete backup system
Todd B. Marshall, 1543 Green Oak Place, Suite 102, Kingwood, TX 77339 (713/358-1800).

WANTED: Anyone with extra issues of Access (except Vol #1, #s 1 & 2) and issues of SOLUS News from 1977 & 1978, please send expected prices to,

C. Jason Slade,
32 Berkley Cres.,
Simcoe, Ontario, CAN.
N3Y 2K4

FOR SALE OR TRADE New North Star double density disk controller board, factory assembled $300.00. New RAM-16, factory assembled $350.00. These were bought for a friend in Japan who reneged on the deal. I also have a used single density controller for $100.00. What do you have to trade? I need a set of Helios controller boards, a bipolar ROM programmer and an inexpensive printer. Joe Maguire, P.O. Box 3742 DT, Anchorage, AK 99510.

HELP WANTED! Want programs for running Houston Instrument Hi-Plot Plotter. Will buy or exchange programs. Write or call collect: Bill Graham, 1937 Rickover Court, New Port Richey, FL 33552 ; (813) 849 7636 evenings.

FREE! PTC Extended cassette basic, PTC 8080 Chess and PTC TREK-80 - with purchase of a working Sol-20 that has 48K memory, complete with video and manuel - asking $1250.

Gene McLaughlin
P O Box 847
Ralls, Texas 79357
Phone - Home (806) 253-2557
- Business 253-2511

Armstrong Corporation

125-F Melody Lane (P.O. Box 533), Costa Mesa, CA 92627 - (714) 752-8050

January 29, 1980

Stanley M. Sokolow
PROTEUS / NEWS
1690 Woodside Road Suite 219
Redwood City, California 94061

Dear Stan;

SOL owners with North Star disk systems might enjoy the following game program. It is from Armstrong Corporation's SOL*STAR game disk #2, and may be reproduced in PROTEUS with their permission. (Game Disk #2 is available for $35 from Armstrong Corp., P.O. Box 10533, Costa Mesa, Calif. 92627.) With only minor modifications, the program will also run using Processor Technology's cassette and disk BASICs.

The program, called STARWARS, is accompanied by another program called STR-RULS, which provides complete documentation for the game. The game simulates the cockpit of a spaceship which you will pilot on intergalactic missions involving face-to-face combat situations with a very aggressive enemy. The pilot who can prevail is accorded honors in a special commendation ceremony. There are nine skill levels in the game. Start with level 7 or 8 to get the feel of things, then try to work yourself up to level 1 -- though you will not think so at first, it is possible to master the highest level!

A final note: Armstrong Corporation's SOL*STAR library contains nearly 100 programs of business and recreational application. I am promised that many can be released for publication in PROTEUS. Therefore, I hope to be a regular contributor in the future. If there are any applications areas that are of special interest to PROTEUS subscribers, please let me know.

Sincerely,

Wm. David Armstrong

Wm. David Armstrong

P.S. Mailing address for PROTEUS subscription:
125-F Melody Lane
Costa Mesa, California 92627

```
****************************************************************
         S T A R W A R S   G A M E   P R O G R A M
****************************************************************

10 REMEMBER TO CREDIT WM. DAVID ARMSTRONG - (C)1981
20 DIM L(28),A$(28),B$(28),C$(28),D$(28)
30 PRINT CHR$(11)
40 PRINT "ENTER SPEED (1=FASTEST, 9=SLOWEST):",
50 W$=INCHAR$(0) \ W=ASC(W$)
60 IF W<49 OR W>57 THEN 50
70 PRINT W$
80 PRINT \ W=W-48
90 S=EXAM(51455)
100 IF S=6 THEN 230
110 Y=51456
120 FOR I=Y TO Y+83
130 READ Z
140 FILL I,Z
150 NEXT I
160 FILL 51455,6
170 PRINT "NEED INSTRUCTIONS?",
180 N=INP(252)
190 IF N=78 THEN 220
200 IF N<>89 THEN 180
205 PRINT " YES"
210 CHAIN "STR-RULS"
220 PRINT " NO" \ PRINT
230 PRINT "PRESS `0' TO START: ",
240 J1=INP(252)
250 IF J1<>48 THEN 240
260 R=5 \ PRINT CHR$(11)
270 A$="TARGET LOCKED IN ON COMPUTER"
280 B$="- NO DIRECT HIT ON TARGET - "
290 D$="  ENEMY PHASER ATTACK !!!!  "
300 D=52224+INT(896*RND(0))
310 GOSUB 650
320 GOSUB 990
330 D=D+J
340 X=CALL(51456,D)
350 FOR I=1 TO W*50 \ NEXT
360 GOTO 500
370 C$=B$
380 L2=53136
390 FOR L1=1 TO 28
400 L(L1)=ASC(C$(L1,L1))-32
410 L3=EXAM(L2+L1)
420 FILL L2+L1,L(L1)+L3
430 NEXT L1
440 FOR I=1 TO W*50 \ NEXT I
450 FOR L1=1 TO 28
460 L3=EXAM(L2+L1)
470 FILL L2+L1,L3-L(L1)
480 NEXT L1
490 RETURN
500 J1=INP(252)
510 C$=A$
520 IF J1>48 AND J1<59 THEN 550
530 GOSUB 380
540 GOTO 500
550 IF J1<>53 THEN 580
560 GOSUB 880
570 GOTO 1170
580 J1=J1-48
585 IF D>51967 AND D<53247 THEN 590
586 D=52571
587 X=CALL(51456,D)
590 IF D<52570 OR D>52572 THEN 610
600 GOTO 1360
```

```
610 GOSUB 1000
620 X=CALL(51456,D)
630 D=D+J
640 GOTO 320
650 PRINT TAB(11)," - - - - - - - - -| | |- - - - - - - - - "
660 PRINT TAB(11),"|                ---                  |"
670 PRINT TAB(11),"|                ---                  |"
680 PRINT TAB(11),"|                ---                  |"
690 PRINT TAB(11),"|          +     ---       +          |"
700 PRINT TAB(11),"--               ---                 --"
710 PRINT "Level",W,
720 PRINT TAB(11),"-- |  |  |  |  |  |   |  |  |  |  |  | --"
730 PRINT TAB(11),"--               ---                 --"
740 PRINT TAB(11),"|          +     ---       +          |"
750 PRINT TAB(11),"|                ---                  |"
760 PRINT TAB(11),"|                ---                  |"
770 PRINT TAB(11),"|                ---                  |"
780 PRINT TAB(11)," - - - - - - - - -| | |- - - - - - - - -"
790 FILL 53120,32
800 RETURN
810 DATA 235,1,39,201,22,5,213,205,26,201
820 DATA 209,21,200,35,3,62,55,133,111,210
830 DATA 6,201,36,195,6,201,30,9,86,10
840 DATA 130,119,29,200,35,3,195,28,201
850 DATA 0,0,128,0,0,0,128,0,0,0,128,0,0,128,0,0,128,0
860 DATA 128,0,128,128,128,128,128,0,128,0,128,0,0,128
870 DATA 0,0,128,0,0,0,128,128,0,0,0,128,0,0
880 H=53184 \ H4=53247
890 FOR H1=1 TO 9
900 H2=EXAM(H) \ H3=EXAM(H4)
910 FILL H,H2+128 \ FILL H4,H3+128
920 H=H-60 \ H4=H4-68
930 NEXT H1
940 IF H6>0 THEN 970
950 H6=1
960 GOTO 880
970 H6=0
980 RETURN
990 J1=INT(9*RND(0)+1)
1000 IF J1>1 THEN 1020
1010 J=-62 \ RETURN
1020 IF J1>2 THEN 1040
1030 J=-64 \ RETURN
1040 IF J1>3 THEN 1060
1050 J=-66 \ RETURN
1060 IF J1>4 THEN 1080
1070 J=2 \ RETURN
1080 IF J1>5 THEN 1100
1090 J=0 \ RETURN
1100 IF J1>6 THEN 1120
1110 J=-2 \ RETURN
1120 IF J1>7 THEN 1140
1130 J=66 \ RETURN
1140 IF J1>8 THEN 1160
1150 J=64 \ RETURN
1160 J=62 \ RETURN
1170 REM SCORE FOR HIT
1180 M1=52702 \ M2=0
1190 M=EXAM(M1+M2)
1200 IF M>128 THEN 1230
1210 GOSUB 370
1220 GOTO 610
1230 M2=M2+1
1240 IF M2<3 THEN 1190
1250 PRINT TAB(12),"D I R E C T   H I T  --  ENEMY DESTROYED"
1260 D=D+65
1270 X=CALL(51456,D)
1280 R=R-1
1290 PRINT TAB(2),"Sensors detect",R," more enemy ships...Press ",
```

```
1300 PRINT "`0' to continue:",
1310 N=INP(252)
1320 IF N<>48 THEN 1310
1330 PRINT CHR$(11)
1340 IF R<1 THEN 1610
1350 GOTO 270
1360 FOR L1=1 TO 28
1370 L(L1)=ASC(D$(L1,L1))
1380 FILL L2+L1,L(L1)
1390 NEXT L1
1400 O=52703
1410 FOR I=1 TO 9
1420 P(I)=0
1430 NEXT I
1440 FOR Q=1 TO 9
1450 FOR Q1=1 TO 9
1460 J1=Q1
1470 GOSUB 1000
1480 P(Q1)=P(Q1)+J
1490 FILL P(Q1),127
1500 NEXT Q1
1510 NEXT Q
1520 PRINT CHR$(11)
1530 FOR I=1 TO 6 \ PRINT \ NEXT I
1540 PRINT TAB(15),"YOU HAVE BEEN DESTROYED BY THE ENEMY!"
1550 FOR I=1 TO 4 \ PRINT \ NEXT I
1560 PRINT TAB(19),"Do you wish to try again? ",
1570 N=INP(252)
1580 IF N=89 THEN 30
1590 IF N<>78 THEN 1570
1600 GOTO 1770
1610 FOR I=1 TO 63 \ PRINT "*", \ NEXT I \ PRINT
1620 PRINT \ PRINT TAB(13),"COMMENDATION FOR MERITORIOUS SERVICE"
1630 PRINT \ FOR I=1 TO 63 \ PRINT "*", \ NEXT I \ PRINT
1640 PRINT \ PRINT "    IN RECOGNITION of exceptional heroism and",
1650 PRINT " performance of"
1660 PRINT "duty in the face of overwhelming odds, the ALLIANCE",
1670 PRINT " hereby"
1680 PRINT "awards its highest honor - THE GALAXY CROSS - and all"
1690 PRINT "rights and privileges appendant, inter alia.",
1700 PRINT "  Level",W," pilot."
1710 PRINT \ FOR I=1 TO 63 \ PRINT "*", \ NEXT \ PRINT \ PRINT
1720PRINT "The Alliance requires your services for yet another "
1730 PRINT "mission.  Will you go? ",
1740 N=INP(252)
1750 IF N=89 THEN 30
1760 IF N<>78 THEN 1740
1770 PRINT CHR$(11)
1780 PRINT "Farewell."
1790 FILL 51455,15
1800 END
```

LOW-COST OPTION GIVES SIX COLORS TO HOUSTON INSTRUMENT PLOTTERS

Now here's a smart idea. Houston Instrument, a division of Bausch & Lomb, has made a name for itself in the microcomputer field by producing a family of low-cost digital plotters at a reasonble price. At the West Coast Computer Faire, they will show a simple retrofittable option that lets the plotter pick up pens of different colors from a parking area at the edge of the plotting bed. Apparently the software can direct the pen carrier to jog on over to the edge of the plotting field, deposit its pen in that pen's home, back away, and go over to another pen's parking stall.

The family of plotters comes with various amounts of intelligence, starting at $1085 for the one-color and $395 more for the 6-color version. Updating of existing units requires $395 plus $100 service charge, plus shipping and all that.

See them at Booth 1226 of the Faire. (Proteus will be in Booth 1448.)

```
***************************************************************
          STR-RULS GAME PROGRAM
***************************************************************

10 REMEMBER TO CREDIT WM. DAVID ARMSTRONG - (C)1981
20 PRINT CHR$(11)
30 PRINT \ PRINT "IF A PRINTER IS CONNECTED TO THIS COMPUTER YOU MAY"
40 PRINT "OBTAIN A HARDCOPY PRINTOUT.  THE FOLLOWING OPTIONS ARE"
50 PRINT "AVAILABILE:" \ PRINT
60 PRINT "0 - OUTPUT TO VIDEO SCREEN ONLY"
70 PRINT "1 - SERIAL PORT OUTPUT"
80 PRINT "2 - PARALLEL PORT OUTPUT"
90 PRINT \ PRINT
100 PRINT "WHICH PRINT OPTION DO YOU SELECT?  ",
110 Z$=INCHAR$(0) \ IF ASC(Z$)<48 OR ASC(Z$)>50 THEN 20
120 Z=VAL(Z$)
130 PRINT CHR$(11)
140 FOR I=1 TO 63 \ PRINT #Z,"*", \ NEXT I \ PRINT #Z
150 PRINT #Z \ PRINT #Z,TAB(18),"RULES FOR STARWARS GAME" \ PRINT #Z
160 FOR I=1 TO 63 \ PRINT #Z,"*", \ NEXT I \ PRINT #Z
170 PRINT #Z \ PRINT #Z
180 PRINT #Z,"A long time ago in a galaxy far, far away...."
190 PRINT #Z
210 FILL 51211,20 \ REM  SLOW DOWN VIDEO DISPLAY
220 PRINT #Z,"It is a period of civil war.  Rebel spaceships,",
230 PRINT #Z," striking from"
240 PRINT #Z,"a hidden base, have won their first victory",
250 PRINT #Z," against the evil"
260 PRINT #Z,"Galactic Empire." \ PRINT #Z
270 PRINT #Z,"During the battle, rebel spies managed to steal",
280 PRINT #Z," secret plans"
290 PRINT #Z,"to the Empire's ultimate weapon, the DEATH STAR,",
300 PRINT #Z," an armored"
310 PRINT #Z,"space station with enough power to destroy an",
320 PRINT #Z," entire planet." \ PRINT #Z
330 PRINT #Z,"Pursued by the Empire's sinister agents, Princess",
340 PRINT #Z," Leia races"
350 PRINT #Z,"home aboard her starship, custodian of the stolen",
360 PRINT #Z," plans that"
370 PRINT #Z,"can save her people and restore freedom to the",
380 PRINT #Z," galaxy...."
390 PRINT #Z \ PRINT #Z,TAB(26),"* * *" \ PRINT #Z
400 PRINT #Z,"So begins 20th Century Fox's popular movie - ",
410 PRINT #Z,"STARWARS.  In this"
420 PRINT #Z,"simulation, you will pilot a rebel spaceship on ",
430 PRINT #Z,"missions which"
440 PRINT #Z,"will bring you into face to face combat with the",
450 PRINT #Z," Empire's"
460 PRINT #Z,"deadly TIE fighters." \ PRINT #Z
470 PRINT #Z,"All flight and fire control for your ship is ",
480 PRINT #Z,"handled via the"
490 PRINT #Z,"10-key pad on your computer console.  The numeral",
500 PRINT #Z," five key, at"
510 PRINT #Z,"the center of your keypad, is the firing button for",
520 PRINT #Z," your"
530 PRINT #Z,"weapons systems.  Motion of your ship is controlled",
540 PRINT #Z," by the"
550 PRINT #Z,"remaining numerical keys - eight is up, two is down,",
560 PRINT #Z," four is"
570 PRINT #Z,"to the left, six is to the right, etc.  The numeral",
580 PRINT #Z," zero key,"
590 PRINT #Z,"when pressed, locks in computer tracking, which has",
600 PRINT #Z," the effect"
605 PRINT #Z,"of freezing the enemy's position on your target grid."
610 PRINT #Z \ PRINT #Z,"The image of an enemy TIE fighter on your ",
620 PRINT #Z,"target grid is the"
630 PRINT #Z,"result of computer sensor scan, and is updated ",
640 PRINT #Z,"approximately"
650 PRINT #Z,"three times per second.  Your opportunity to move ",
660 PRINT #Z,"or fire"
670 PRINT #Z,"occurs for a brief period at the completion of each",
680 PRINT #Z," scan.  In"
690 PRINT #Z,"the absence of a new move or command, your last ",
700 PRINT #Z,"command will"
710 PRINT #Z,"be interpreted as your current command." \ PRINT #Z
720 PRINT #Z,"The enemy can only be destroyed by a direct hit ",
730 PRINT #Z,"(when the"
740 PRINT #Z,"center of the target grid is completely filled by",
750 PRINT #Z," the target)."
760 PRINT #Z,"In addition, when the enemy is in firing position",
770 PRINT #Z," on your"
780 PRINT #Z,"target grid, you are also in firing position on his",
790 PRINT #Z," target"
800 PRINT #Z,"grid.  In the absence of an offensive reaction ",
810 PRINT #Z,"on your part,"
820 PRINT #Z,"the enemy will automatically fire upon you and ",
830 PRINT #Z,"destroy you!" \ PRINT #Z
840 PRINT #Z,"A few final points.... Do not try to control the ",
850 PRINT #Z,"enemy, fly"
860 PRINT #Z,"toward him with your ship.  He will be making evasive"
870 PRINT #Z,"maneuvers.  If you let the enemy escape off of your",
880 PRINT #Z," target"
890 PRINT #Z,"grid, you may expect him to reappear with a surprise",
900 PRINT #Z," attack." \ PRINT #Z
910 PRINT #Z,"May the Force be with you."
920 FILL 51211,00
930 FOR I=1 TO 2500 \ NEXT
940 PRINT \ PRINT \ PRINT "THE GAME WILL NOW COMMENCE"
950 FOR I=1 TO 1000 \ NEXT
955 FILL 51455,6
960 CHAIN"STARWARS"
970 END
```

February 15, 1981

Dear Stan,

The routine for Solos is fairly easy to implement. Just type in the code at C9ØØ as shown and enter the custom command as per the instructions in the Solos manual. Decimal arguments cannot be used with this version since there are no decimal conversion routines within Solos.

The version for the North Star Monitor 5.2 requires more extensive patching. The Monitor program occupies the full 8 blocks in which it is stored on the disk so a new file of 1Ø blocks (double density requires an even number of blocks) must be created to hold the Monitor and the Hex Math routine. Proceed as follows:

```
CR NEWMON 1Ø            Create the new file of length 1Ø blocks.
TY NEWMON 1 EØØ         Make it type 1 with a GO address of EØØ.
LF MØEØØ 1EØØ           Load the Monitor with an offset of 1ØØØH.
GO MØEØØ                Execute another copy of the Monitor.
```

Using the DS command of the Monitor, patch in the code given in the listing at the address given plus the offset used in loading the copy to be patched. For example, to enter the new command in the command table type DS 2Ø5E <CR> and then the code as per instructions given in the North Star Software manual. After the patching has been completed, then save the new monitor.

```
SF NEWMON 1EØØ          Save the patched copy of MØEØØ in NEWMON.
GO NEWMON               Execute the patched copy.
HM value value          Try the new command.
```

All the rules applicable to arguments used with other Monitor commands apply to the new HM command including decimal values (followed by a T) and editing functions. This is an advantage of using subroutines already within the program (if you can find them) rather than writing your own. If you do not have the version 5.0 or later software from North Star, (5.2 is the latest) you can try searching for the required subroutines in older versions which should be somewhere near the addresses used in releases 5.0 and above. The complete release of 5.2 software from North Star can be ordered from the North Star Users Association for $25.00 which includes a one year subscription to their newsletter. The address is:

INSUA
P.O. Box 1318,
Antioch, CA 94509

Sincerely, Joe Maguire

P.S. The extra space left over in those two blocks added to NEWMON will not go to waste. I have an improved memory search routine which is almost ready. The improvement will be a wild card option which will allow ambiguous blocks of code to be found. For example, to find all calls to the C4 address area of Solos, you will be able to type: SM CD,*,C4 and the "*" will match any byte value. This feature is already available in my program MEDIT on library disk H6. How do you think I found the required subroutines in Monitor 5.2?

```
                            *
                            *   Hex Math routine for Solos monitor
                            *
                            *   Syntax:  HM  value1  value2
                            *            Values can be anything from 0 to FFFFH
                            *            Only legal Hex permitted.  Overflow or
                            *            underflow is ignored.  Both values must
                            *            be present.
                            *
                            *   Subroutine addresses within Solos monitor
                            *
     C019                   SOUT    EQU     0C019H  Output a character from B
     C33A                   SCONV   EQU     0C33AH  Get a 16 bit value to HL
     C3E8                   ADOUT   EQU     0C3E8H  Output a 16 bit value from HL
     C56A                   NLOOP   EQU     0C56AH  Output a string pointed to by HL
                            *
C83C                        CUTAB   ORG     0C83CH  Custom command table entry
                            *
C83C 48 4D                          ASC     'HM'    Command syntax
C83E 00 C9                          DW      HMATH
C840 00                             DB      0       End of table mark
                            *
C900                        HMATH   ORG     0C900H  Hex Math routine address
                            *
C900 CD 3A C3                       CALL    SCONV   Get value1, abort if missing
C903 E5                             PUSH    H       Save it
C904 CD 3A C3                       CALL    SCONV   Get value2, abort if missing
C907 E5                             PUSH    H       Save it
C908 21 28 C9                       LXI     H,MSG   Point to format string
C90B 16 06                          MVI     D,6     String counter
C90D CD 6A C5                       CALL    NLOOP   Send it out
C910 D1                             POP     D       Get value2 to DE
C911 E1                             POP     H       Get value1 to HL
C912 E5                             PUSH    H       Save it again
C913 19                             DAD     D       Add DE to HL
C914 CD E8 C3                       CALL    ADOUT   Send out the result
C917 E1                             POP     H       Get value1 back to HL
                            *
C918 7A                             MOV     A,D     Now get two's complement
C919 2F                             CMA     .       of DE to find difference
C91A 57                             MOV     D,A
C91B 7B                             MOV     A,E
C91C 2F                             CMA     .
C91D 5F                             MOV     E,A
C91E 13                             INX     D
                            *
C91F 19                             DAD     D       Add it to HL
C920 06 2D                          MVI     B,'-'   Send out minus sign
C922 CD 19 C0                       CALL    SOUT
C925 C3 E8 C3                       JMP     ADOUT   Send out minus result
                            *
C928 20 20 3D 20            MSG:    ASC     '  =  +'
     20 2B
                            *
                                    END
```

```
                 ;
                 ; Hex Math routine for NorthStar Monitor 5.2
                 ;
                 ; Syntax:  HM  value1  value2
                 ;          Values can be anything from 0 to FFFFH
                 ;          or 0 to 65535T (decimal).  Overflow or
                 ;          underflow is ignored.  Both values must
                 ;          be present.
                 ;
0E00             ENTRY:  ORG  0E00H  ;For Monitor at 0E00H as supplied
                 ;                    on factory master disk.  You can
                 ;                    change this as required.
                 ;
105E             HMCOM:  ORG  ENTRY+25EH  ;Command table entry
                 ;
105E 484D                DB      'HM'    ;Command syntax
1060 0016                DW      HMATH
1062 00                  DB      0       ;End of table mark
                 ;
                 ; Subroutine addresses within Monitor 5.2
                 ;
1181 =           VOUT:   EQU  ENTRY+381H  ;Output a 16 bit value from HL
12D7 =           COUT:   EQU  ENTRY+4D7H  ;Output a character from A
12E7 =           SOUT:   EQU  ENTRY+4E7H  ;Output a string pointed to by HL
12F1 =           CRLF:   EQU  ENTRY+4F1H  ;Output C/R and L/F
147D =           GETV:   EQU  ENTRY+67DH  ;Get a 16 bit value to HL
                 ;
1600             HMATH:  ORG  ENTRY+800H  ;Hex Math routine address
                 ;
1600 CD7D14              CALL    GETV    ;Get value1
1603 E5                  PUSH    H       ;Save it
1604 CD7D14              CALL    GETV    ;Get value2
1607 E5                  PUSH    H       ;Save it
1608 212916              LXI     H,MSG   ;Point to format string
160B CDE712              CALL    SOUT    ;Send it out
160E D1                  POP     D       ;Get value2 to DE
160F E1                  POP     H       ;Get value1 to HL
1610 E5                  PUSH    H       ;Save it again
1611 19                  DAD     D       ;Add DE to HL
1612 CD8111              CALL    VOUT    ;Send out the result
1615 E1                  POP     H       ;Get value1 back to HL
                 ;
1616 7A                  MOV     A,D     ;Now get two's complement
1617 2F                  CMA             ;of DE to find difference
1618 57                  MOV     D,A
1619 7B                  MOV     A,E
161A 2F                  CMA
161B 5F                  MOV     E,A
161C 13                  INX     D
                 ;
161D 19                  DAD     D       ;Add it to HL
161E 3E2D                MVI     A,'-'   ;Send out minus sign
1620 CDD712              CALL    COUT
1623 CD8111              CALL    VOUT    ;Send out minus result
1626 C3F112              JMP     CRLF    ;Give a C/R & L/F & return
                 ;
1629 20203D      MSG:    DB      '  ='
162C 20202B              DB      '  +'
162F 00                  DB      0       ;End of message mark
                 ;
1630                     END
```

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

USEFUL ROUTINES FOR INTERFACING THE SOLOS VDM DRIVER AND MICROPOLIS DISK BASIC

By Melvin M. Dalton

INTRODUCTION

SOLOS provides for many different ways of handling data to or from the screen. PTCo's BASIC5 and ECBASIC have built in functions which take advantage of the SOLOS/CUTER/VDM versatility. Other BASIC's do not have these functions built in for obvious reasons. Since many of us use other BASICs (especially disk BASICs), I have designed the following subroutines to be called as needed as assembly language functions from the BASIC program.

These programs have been prepared to function with the MICROPOLIS BASIC, but should provide some useful ideas to those who use similar yet different BASICs.

The assembly language program has been located in the SOLOS RAM area using a program origin of C980. It can be moved into system RAM by assembling the source with a different origin. If this is done, be SURE to include in any BASIC programs the nescesary MEMEND statement to reserve room for the assembly language routines.

The program listing appears after this article along with a sample BASIC program to illustrate the use of the various routines. Each listing is heavily commented so a detailed discussion of them will not be provided. These programs have been sent to the Proteus Cassette Librarian for non-exclusive distribution to Proteus members.

DESCRIPTION

A. Control Character Routine

1. Function:-- Each character in the input string is converted to the corresponding control code and sent to the VDM driver.
2. Syntax:-- <string variable>=FA<letter> (<input string>)
   i.e. Z$=FAC("[AZ") or F$=FAA(C$)
3. Entry point:-- C9A1
4. Error trapping:-- The string variable will be assigned SYNTAX ERROR C if the input argument is not a string. Otherwise the string variable will be the null string.
5. Uses:-- All of the control character and escape code sequences described in the SOLOS manual on pages 18-19 can be generated singly or in groups.
   Examples: Z$=FAC("K") will clear screen & home cursor.
   Z$=FAC("[A@M") clears all of the current line.
   NOTE: Very handy as a substitute for the "&" function of PTCo ECBASIC.

B. Delay While Fetch Routine

1. Function:-- The keyboard is checked for activity while a delay proportional to the input variable elapse. When a key is struck, then routine returns byte immediately. Otherwise a null string is returned at the end of the delay.
2. Syntax:-- <string variable>=FA<letter> (<argument1>)
   where <argument1> is a whole real number from 0 to 99.
   i.e. Z$=FAD(45) or D$=FAW(D)
3. Entry point:-- C9C5
4. Error trapping:-- The string variable will contain SYNTAX ERROR D if argument1 is a string or is defined as an integer (i. e. D%=45 is an integer while D=45 is a real). The same message is returned if argument1 is negative or greater than 99.
5. Uses:-- Provides a real time response requirement in BASIC games. The delay can be shortened to increase difficulty or speed of play. An INPUT statement gives an infinite time to play while sampling the input port with an INP(port number) statement gives practically no time for response.

C. Inverse Video Routine

1. Function:-- Each character in the input string is inverted (changed to black on white) and sent to the VDM driver.
2. Syntax:-- <string variable>=FA<letter> (<input string>)
   i.e. Z$=FAI("INVERSE") or I$=FAA(V$)
3. Entry point:-- C9F5
4. Error trapping:-- The string variable will be assigned SYNTAX ERROR I if the input argument is not a string. Otherwise the string variable will be the null string.
5. Uses:-- Very handy for highlighting prompt or error messages. Also spruces up game displays.

D. Terminal Mode Routine

1. Function:-- Each character entered on the keyboard is echoed to the screen unchanged. Striking the ESCAPE key returns program control to the BASIC program.
2. Syntax:-- <string variable>=FA<letter>
   i.e. Z$=FAT or T$=FAQ
3. Entry point:-- C980
4. Error trapping:-- The string variable will be assigned SYNTAX ERROR T if there is any argument of any type. Otherwise the string variable will be the null string.
5. Uses:-- As a poor man's text editor on one screen's worth of data. SOL cursur control keys will move cursor anywhere needed to make correction or addition. Use Text Line Return Routine (E. below) to send back to BASIC any line or part of line desired.

E. Text Line Return Routine

1. Function:-- A string variable is returned which has the data currently on the screen on the line specified starting with the column specified and continuing to the end of that line. Leading and imbedded blanks are included but trailing blanks are not.
2. Syntax:-- <string variable>=FA<letter>
   (<argument1>,<argument2>)
   where: argument1=line number(0=top,15=bottom)
   argument2=column number(0=left edge,
   63=right edge)
   i.e. Z$=FAL(6,28) or S$=FAB(R,C)
3. Entry point:-- CA2A
4. Error trapping:-- The string variable will be assigned SYNTAX ERROR L if there are not TWO arguments, or if either or both arguments are not real whole numbers. Argument1 must be in the range 0-15 while argument2 must range from 0-63 or the error message is returned. Otherwise the string variable will be the null string.
5. Uses:-- A multi-line mailing label, invoice, etc. can be displayed on the screen at one time using the Control Character Routine and PRINT statements. The Terminal Mode Routine can then be used to enter or edit the data. Then repeated use of this routine will recover the correct data for use by the BASIC program.

COMMENTS

1. The Control Character Routine is limited in its ability to place the cursor. The largest control character is 1F hex or 31 decimal which when used in SOLOS escape sequence 1B 01 ## (see page 18 of SOLOS manual) means the cursor is set to the middle of the line. To reach positions in the right half of the screen use the Text Line Return Routine. Simply set the two arguments to the desired row and column and after calling the routine just ignore the string expression that is returned.
2. If the cursor is ever placed where inverse video is displayed, and then moved, it will leave an un-inverted character behind.
3. Be sure to use the same third letter in each call for a particular function as you used in the DEF statement which defined its entry point.

4. Line number references used in the Control Character Routine (escape sequence 1B 01 ##) are to absolute line numbers (i.e. 0 is always the top line on the screen and 15 is always the bottom line). If scrolling takes place between references, then the line number refered to must be adjusted to the new physical location on the screen of the data you want. The same applies to line references used in the Text Line Return Routine.

PROGRAM LISTING

```
0000                *
0000                *
0000                ***** SPECIAL UTILITIES FOR USE BY
0000                ***** MICROPOLIS BASIC PROGRAMS.
0000                *
0000                ***** BY MELVIN M. DALTON
0000                *****   NON-EXCLUSIVE RIGHTS DONATED TO
0000                *****   PROTEUS CASSETTE LIBRARY
0000                *****   DATED 1/30/81
0000                *
0000                          TAB     10,17,28
0000      C01F      SINP      EQU     0C01FH
0000      C019      SOUT      EQU     0C019H
0000      01A0      RESULT    EQU     1A0H
0000      04BC      ARG1      EQU     4BCH
0000      04BE      ARG2      EQU     ARG1+2
0000      04C4      NARGS     EQU     ARG1+8
0000      04C5      RSIZE     EQU     ARG1+9
0000                          ORG     0C980H        ;SOL RAM
C980                *
C980                *   *****TERMINAL MODE*****
C980                *
C980                *   KEYBOARD GOES DIRECTLY TO SCREEN
C980                *     EXIT=ESC KEY
C980                *
C980 3E 54          TERMINAL  MVI     A,'T'         ;ERROR CODE
C982 CD 8B CA                 CALL    SYNERR        ;SET UP RET MSG
C985                                                ;IN CASE OF ERROR
C985 3A C4 04                 LDA     NARGS         ;# OF ARGUMENTS
C988 A7                       ANA     A             ;SET FLAGS
C989 C0                       RNZ                   ;ERROR IF NOT 0
C98A CD 1F C0       TERM1     CALL    SINP          ;GET CHARACTER
C98D CA 8A C9                 JZ      TERM1
C990 FE 1B                    CPI     1BH           ;ESC KEY?
C992 CA 9C C9                 JZ      BASRET        ;YES, RETURN TO BASIC
C995 47                       MOV     B,A
C996 CD 19 C0                 CALL    SOUT          ;PUT ON SCREEN
C999 C3 8A C9                 JMP     TERM1         ;REPEAT
C99C AF             BASRET    XRA     A             ;GET ZERO
C99D 32 A2 01                 STA     RESULT+2      ;MAKE RET MSG
C9A0                                                ;EQUAL ZERO LENGTH
C9A0 C9                       RET                   ;BACK TO BASIC
C9A1                *
C9A1                *   *****CONTROL CHARACTER MODE*****
C9A1                *
C9A1                *   CONVERT ARGUMENT STRING TO CTRL
C9A1                *   CHARACTERS AND SEND RESULTS TO SCREEN
C9A1                *
C9A1 3E 43          CONTROL   MVI     A,'C'         ;ERROR CODE
C9A3 CD 8B CA                 CALL    SYNERR        ;SET UP ERR RESULT
C9A6 3A C4 04                 LDA     NARGS         ;# OF ARGUMENTS
C9A9 FE 01                    CPI     1             ;IF NOT 1
C9AB C0                       RNZ                   ;SYNTAX ERROR
C9AC 2A BC 04                 LHLD    ARG1          ;POINT TO ARGUMENT
C9AF 7E                       MOV     A,M           ;GET TYPE
C9B0 FE 03                    CPI     3             ;STRING?
C9B2 C0                       RNZ                   ;SYNTAX ERROR
C9B3 23                       INX     H
```

```
C9B4 23                       INX     H             ;POINT TO STRING LENGTH
C9B5 4E                       MOV     C,M           ;COUNT IN "C"
C9B6 23             CNTL1     INX     H             ;1ST CHARACTER
C9B7 7E                       MOV     A,M
C9B8 E6 1F                    ANI     01FH          ;STRIP TO CTRL CHAR.
C9BA 47                       MOV     B,A
C9BB CD 19 C0                 CALL    SOUT          ;TO SCREEN
C9BE 0D                       DCR     C             ;ONE LESS CHAR.
C9BF C2 B6 C9                 JNZ     CNTL1
C9C2 C3 9C C9                 JMP     BASRET        ;BACK TO BASIC
C9C5                *
C9C5                *   *****DELAY WHILE FETCH MODE*****
C9C5                *
C9C5                *   DELAY TIME SPECIFIED IN ARGUMENT
C9C5                *   0=MINIMUM DELAY, 99=MAXIMUM DELAY
C9C5                *   RETURN BYTE ANYTIME KEY STRUCK
C9C5                *   OTHERWISE RETURN NULL STRING
C9C5                *
C9C5 3E 44          DELAY     MVI     A,'D'         ;ERROR CODE
C9C7 CD 8B CA                 CALL    SYNERR
C9CA 3A C4 04                 LDA     NARGS
C9CD FE 01                    CPI     1
C9CF C0                       RNZ
C9D0 2A BC 04                 LHLD    ARG1
C9D3 7E                       MOV     A,M           ;ARGUMENT TYPE
C9D4 FE 01                    CPI     1             ;REAL?
C9D6 C0                       RNZ
C9D7 CD B3 CA                 CALL    BCDHEXR       ;CONVERT TWO LSB TO HEX
C9DA 47                       MOV     B,A           ;COUNT IN "BC"
C9DB 0E FF                    MVI     C,0FFH        ;COUNT TIMES 256
C9DD                                                ;MIN. COUNT=255
C9DD CD 1F C0       DLY1      CALL    SINP          ;CHECK KBD
C9E0 32 A3 01                 STA     RESULT+3      ;DATA BYTE
C9E3 C2 EF C9                 JNZ     BYTBAS        ;RET BYTE TO BASIC
C9E6 0B                       DCX     B             ;COUNT
C9E7 78                       MOV     A,B           ;RET TO BASIC IF
C9E8 B1                       ORA     C             ;DELAY EXPIRED
C9E9 C2 DD C9                 JNZ     DLY1          ;REPEAT
C9EC C3 9C C9                 JMP     BASRET        ;ZERO LENGTH RET
C9EF 3E 01          BYTBAS    MVI     A,1           ;STRING LENGTH
C9F1 32 A2 01                 STA     RESULT+2      ;IN BUFFER
C9F4 C9                       RET                   ;NOW RETURN
C9F5                *
C9F5                *   *****INVERSE VIDEO MODE*****
C9F5                *
C9F5                *   ARGUMENT STRING IS OUTPUT
C9F5                *   AS INVERTED VIDEO
C9F5                *
C9F5 3E 49          INVIDEO   MVI     A,'I'         ;ERROR CODE
C9F7 CD 8B CA                 CALL    SYNERR        ;SET UP ERROR RET
C9FA 3A C4 04                 LDA     NARGS         ;# OF ARGUMENTS
C9FD FE 01                    CPI     1             ;SET FLAGS
C9FF C0                       RNZ                   ;TO BASIC IF WRONG
CA00 2A BC 04                 LHLD    ARG1          ;POINT TO INPUT
CA03 7E                       MOV     A,M           ;GET TYPE BYTE
CA04 FE 03                    CPI     3             ;MUST BE STRING
CA06 C0                       RNZ                   ;ERROR RET
CA07 23                       INX     H
CA08 23                       INX     H             ;LENGTH COUNT
CA09 4E                       MOV     C,M           ;IN "C"
CA0A 23                       INX     H             ;POINT FIRST BYTE
CA0B 3E 07          INVLOOP   MVI     A,7           ;ESC SEQ CODE
CA0D CD 1F CA                 CALL    ESCOUT        ;OUTPUT ESC. SEQ.
CA10 7E                       MOV     A,M           ;GET BYTE TO INVERT
CA11 F6 80                    ORI     80H           ;INVERT IT
CA13 47                       MOV     B,A
CA14 CD 19 C0                 CALL    SOUT          ;OUT TO SCREEN
CA17 23                       INX     H             ;POINT NEXT BYTE
CA18 0D                       DCR     C             ;COUNT DOWN
CA19 C2 0B CA                 JNZ     INVLOOP       ;REPEAT
```

```
CA1C C3 9C C9            JMP     BASRET     ;ZERO LENGTH RET
CA1F            *
CA1F            *   OUTPUT 1BH FOLLOWED BY BYTE IN 'A'
CA1F            *
CA1F F5         ESCOUT   PUSH    PSW        ;SAVE 2ND BYTE
CA20 06 1B               MVI     B,1BH      ;ESCAPE CODE
CA22 CD 19 C0            CALL    SOUT
CA25 F1                  POP     PSW        ;BYTE BACK
CA26 47                  MOV     B,A
CA27 C3 19 C0            JMP     SOUT       ;OUT & RET
CA2A            *
CA2A            *   *****TEXT LINE RETURN MODE*****
CA2A            *
CA2A            *   ON INPUT: SET CURSOR TO SPECIFIED
CA2A            *             ROW AND COLUMN.
CA2A            *   ON RETURN: RETURN CURRENT SCREEN LINE
CA2A            *              CONTENTS FROM CURSOR TO
CA2A            *              LAST NON-BLANK CHARACTER
CA2A            *
CA2A 3E 4C      LNTEXT   MVI     A,'L'      ;ERROR CODE
CA2C CD 8B CA            CALL    SYNERR     ;PRE-SET ERROR
CA2F 3A C4 04            LDA     NARGS
CA32 FE 02               CPI     2          ;SHOULD BE TWO
CA34 C0                  RNZ                ;ERROR
CA35 2A BC 04            LHLD    ARG1
CA38 7E                  MOV     A,M        ;TYPE?
CA39 FE 01               CPI     1          ;REAL?
CA3B C0                  RNZ                ;ERROR
CA3C 3E 02               MVI     A,2        ;PREPARE TO SET CURSOR
CA3E CD 1F CA            CALL    ESCOUT     ;TO DESIRED ROW(LINE)
CA41 CD B3 CA            CALL    BCDHEXR    ;CONVERT TO HEX
CA44 FE 10               CPI     10H        ;MAX ROWS=16
CA46 D0                  RNC                ;ERROR
CA47 47                  MOV     B,A
CA48 CD 19 C0            CALL    SOUT       ;CURSOR TO DESIRED ROW
CA4B 2A BE 04            LHLD    ARG2
CA4E 7E                  MOV     A,M
CA4F FE 01               CPI     1          ;REAL
CA51 C0                  RNZ
CA52 3E 01               MVI     A,1        ;COLUMN CODE
CA54 CD 1F CA            CALL    ESCOUT
CA57 CD B3 CA            CALL    BCDHEXR
CA5A FE 40               CPI     40H        ;MAX COL.=64
CA5C D0                  RNC
CA5D 47                  MOV     B,A
CA5E CD 19 C0            CALL    SOUT       ;CURSOR TO COLUMN
CA61 3E 04               MVI     A,4        ;GET CURSOR SCREEN
CA63                                        ;ADDRESS IN 'BC'
CA63 CD 1F CA            CALL    ESCOUT
CA66 C5                  PUSH    B          ;SAVE ADDRESS
CA67 79                  MOV     A,C
CA68 E6 3F               ANI     3FH        ;CURRENT COLUMN
CA6A 4F                  MOV     C,A
CA6B 3E 40               MVI     A,40H      ;64 COLUMNS
CA6D 91                  SUB     C          ;MAX LINE LENGTH
CA6E 4F                  MOV     C,A        ;COUNT IN C
CA6F E1                  POP     H          ;ADDRESS IN HL
CA70 E5                  PUSH    H          ;RESAVE
CA71 7D                  MOV     A,L
CA72 F6 3F               ORI     3FH
CA74 6F                  MOV     L,A        ;'HL'=END OF LINE
CA75 7E         BLANKS   MOV     A,M        ;GET BYTE
CA76 FE 20               CPI     20H        ;BLANK?
CA78 C2 80 CA            JNZ     BLANK1     ;FOUND END OF LINE
CA7B 2B                  DCX     H          ;NEXT BYTE
CA7C 0D                  DCR     C          ;COUNT DOWN
CA7D C2 75 CA            JNZ     BLANKS     ;LOOP
CA80 79         BLANK1   MOV     A,C        ;LINE LENGTH
CA81 32 A2 01            STA     RESULT+2   ;IN RESULT
CA84 E1                  POP     H          ;START OF LINE
```

```
CA85 11 A3 01            LXI     D,RESULT+3 ;PUT IT HERE
CA88 C3 AA CA            JMP     MOVE       ;MOVE IT
CA8B            *
CA8B            *   INSERT 'SYNTAX ERROR' IN
CA8B            *   'RESULT' BUFFER
CA8B            *
CA8B 32 A9 CA   SYNERR   STA     TYPBYT     ;ERROR CODE
CA8E 21 99 CA            LXI     H,ERRMSG   ;POINT TO MSG
CA91 11 A0 01            LXI     D,RESULT   ;POINT TO BUFFER
CA94 0E 11               MVI     C,17       ;MSG LENGTH
CA96 C3 AA CA            JMP     MOVE       ;MOVE MSG & RET
CA99 03 40 0E   ERRMSG   DB      3,64,14
CA9C 53 59 4E            DT      'SYNTAX ERROR'
CA9F 54 41 58
CAA2 20 45 52
CAA5 52 4F 52
CAA8 20                  DB      20H        ;SPACE
CAA9 00         TYPBYT   DB      0          ;RESERVED FOR TYPE
CAAA            *
CAAA            *   MOVE FROM 'HL' TO 'DE'
CAAA            *   A TOTAL OF 'C' BYTES
CAAA            *
CAAA 7E         MOVE     MOV     A,M        ;GET BYTE
CAAB 12                  STAX    D          ;PUT IN BUFFER
CAAC 23                  INX     H
CAAD 13                  INX     D
CAAE 0D                  DCR     C          ;COUNT BYTES
CAAF C8                  RZ                 ;RET IF DONE
CAB0 C3 AA CA            JMP     MOVE       ;REPEAT IF NOT
CAB3            *
CAB3            *   SUBROUTINE TO CONVERT ONE OR TWO
CAB3            *   DIGITS OF A REAL WHOLE NUMBER
CAB3            *   FROM ARGUMENT FORMAT INTO A HEX BYTE
CAB3            *   ON ENTRY:  HL POINTS TO ARGUMENT
CAB3            *   ON EXIT:   A  CONTAINS HEX VALUE
CAB3            *                 EQUAL TO TWO LSB OF
CAB3            *                 ARGUMENT
CAB3            *
CAB3 23         BCDHEXR  INX     H          ;POINT TO EXPONENT
CAB4 7E                  MOV     A,M
CAB5 A7                  ANA     A          ;ZERO?
CAB6 C8                  RZ                 ;RET WITH 'A'=0
CAB7 23                  INX     H          ;POINT TO DIGIT(S)
CAB8 FE 41               CPI     41H        ;I.E. 1-9
CABA CA D0 CA            JZ      BCDHEXR1
CABD FE 42               CPI     42H        ;I.E. 10-99
CABF C2 D6 CA            JNZ     ERREXIT    ;WRONG SIZE NUMBER
CAC2 7E                  MOV     A,M        ;GET 2 BCD DIGITS
CAC3 E6 0F               ANI     0FH        ;LSB DIGIT
CAC5 5F                  MOV     E,A        ;SAVE IN 'E'
CAC6 7E                  MOV     A,M        ;GET DIGITS BACK
CAC7 E6 F0               ANI     0F0H       ;MSB DIGIT
CAC9 1F                  RAR                ;MSB TIMES 8
CACA 57                  MOV     D,A        ;SAVE IN 'D'
CACB 1F                  RAR
CACC 1F                  RAR                ;MSB TIMES 2
CACD 82                  ADD     D          ;MSB TIMES 10
CACE 83                  ADD     E          ;PLUS LSB
CACF C9                  RET
CAD0 7E         BCDHEXR1 MOV     A,M        ;GET DIGIT
CAD1 1F                  RAR
CAD2 1F                  RAR
CAD3 1F                  RAR
CAD4 1F                  RAR                ;NOW HEX
CAD5 C9                  RET
CAD6 E1         ERREXIT  POP     H          ;REALIGN STACK
CAD7 C9                  RET                ;ALL THE WAY
CAD8                                        ;BACK TO BASIC
CAD8                     END

ERRORS THIS ASSEMBLY  0000
```

SAMPLE PROGRAM

```
10 LOAD 'UTLO': REM LOAD OBJECT CODE OF UTILITIES ROUTINES
20 DEF FAC=16RC9A1: REM ENTRY POINT FOR CTRL CHAR. MODE
30 DEF FAT=16RC980: REM ENTRY POINT FOR TERMINAL MODE
40 DEF FAD=16RC9C5: REM ENTRY POINT FOR DELAY & FETCH MODE
50 DEF FAL=16RCA2A: REM ENTRY POINT FOR TEXT LINE RETURN MODE
60 DEF FAI=16RC9F5: REM ENTRY POINT FOR INVERSE VIDEO MODE
70 REM DIMENSION I$ & Z$ FOR LENGTH OF 64 (DEFAULT=40)
80 DIM I$(5,64),Z$(64)
90 REM NOW CLEAR SCREEN
100 Z$=FAC('K'): IF Z$<>'' THEN PRINT Z$: STOP
110 REM IF-THEN TEST ABOVE WILL PRINT SYNTAX ERROR IF IT OCCURS
120 REM NOW PUT CURSOR ON LINE 5
130 Z$=FAC('[BE'): IF Z$<>'' THEN PRINT Z$: STOP
140 REM LOAD LIST I$ WITH PROMPT LIST
150 FOR I=0 TO 5: READ I$(I): NEXT I
160 DATA 'NAME      :'
170 DATA 'STREET    :'
180 DATA 'CITY      :'
190 DATA 'STATE     :'
200 DATA 'ZIP       :'
210 DATA 'PHONE     :'
220 REM PRINT THE PROMPT LIST
230 FOR I=0 TO 5: PRINT TAB(10);I$(I): NEXT I
240 Z$=FAC('Z[AE'): REM DOWN ONE LINE & OVER FIVE COLUMNS
250 REM USE INVERSE VIDEO TO REQUEST DATA IN TERMINAL MODE
260 Z$=FAI(' ENTER DATA REQUESTED '):IFZ$<>''THENPRINTZ$:STOP
270 REM PUT CURSOR WHERE INPUT DATA SHOULD START
280 Z$=FAL(5,19): IF LEFT$(Z$,6)='SYNTAX' THEN PRINT Z$: STOP
290 REM GO TO TERMINAL MODE
300 Z$=FAT: IF LEN(Z$)<>0 THEN PRINT Z$: STOP
310 REM NOW GET TEXT FROM SCREEN & PUT IN LIST I$
320 FOR I=0 TO 5
330 Z$=FAL(5+I,19): IF LEFT$(Z$,6)='SYNTAX' THEN PRINT Z$: STOP
340 REM NOTE SPECIAL FORM OF ERROR TEST SINCE FAL MAY RETURN
350 REM NULL STRING IF SCREEN HAS ONLY BLANKS FOR A LINE
360 I$(I)=Z$
370 NEXT I
380 REM NOW SKIP LINE, ERASE NEXT LINE, & REPLACE PROMPT
390 Z$=FAC('ZMZMSSSSS'): IF Z$<>'' THEN PRINT Z$: STOP
400 Z$=FAI(' THANK YOU '): IF Z$<>'' THEN PRINT Z$: STOP
410 REM GO TO START OF NEXT LINE AND OUTPUT RESULTS
420 REM USUALLY THE RESULTS WOULD GO TO DISK OR MORE COMPUTING
430 Z$=FAC('MZ'): IF Z$<>'' THEN PRINT Z$: STOP
440 PRINT I$(0)+' , '+I$(1)
450 PRINT I$(2)+',  '+I$(3)+'      '+I$(4)+'      '+I$(5)
460 END
```

BUG SQUAD

Along with the New Year's resolutions came the New Year's bugs! A user of the ALS8 simulator instruction decoder (Proteus News Vol. 3 #1) wrote to say the the RST instructions do not decode properly. Sure enough, all the RSTs decode as RST 20. To fix this one, insert the line: CALL XTRAC between lines 990 and 1000 in the listing as given on page 12. Somehow it got left out.

Joe Maguire

Jan. 25, 1981
813 Inlet Rd.
Green Lake,WI
54941

Dear Stan,

A number of things that I thought interesting have happened to me in the last few months so I report to the members as follows.

METRON COMPUTERWARE INC., 552 W. 114th St., NY, NY 10025 are selling the SOS personality module which relocates SOLOS to F000H. They have run ads in the last two issues of BYTE and by now most members have probably seen the ad. The changes to the SOL are very simple and after they are made you can have either a 48K Sol or a 60K Sol just by interchanging memory and the Personality Modules. I am limited to a 56K system as I use NORTH STAR DD which has a PROM located at E800H. I ordinarily use a P.T. 48KRA board in my machine and I add a P.T. 16K board when I use the SOS module. By accident I discovered that the extra 16K board can stay in the machine even when I use SOLOS at it's normal address of C000H.

The changes to Lifeboats version of CP/M are limited to changing the two addresses in the user area that call SOLOS. Unfortunately no hope is in sight for the use of PTDOS with this module. The METRON people have told me that the re-write job that would be required to PTDOS is just not feasable.

The SOL that I use at home still has both HELIOS and NORTH STAR drives installed. Because of the reliability factor I have changed virtually all of my programs and files over to NORTH STAR and CP/M. In my opinion the PERSCI drives that PT used just had to be a cause of their demise. I use an ALPHA-MICRO as well which started out life with PERSCI drives and later was lucky enough to get a hard disk. The members of the ALPHA-MICRO users group were talking about a class action suit against PERSCI !

The CP/M users group has recently put out their library disks on N*DD. I have recieved all of the volumes which have software that is related to SOL and I am now in the process of going through each volume and sorting out the stuff that PROTEUS members would be interested in. Each volume contains a mixed bag of programs many of which are for specific machines or for Z-80 machines. As soon as I get this job done I will make available on just two or three N* disks the software which I think others may want copies of.

My latest effort includes the use of a SOL as a host for a normal CP/M machine with a terminal attached. In this case I wanted a full 24 X 80 character display. Since the terminal uses the SOL serial port I had to give up my usual DIABLO printer and find a parallel printer fopr my one extra plug on the back of the SOL. I settled on the new MX-80 printer which is very cheap and does a suprisingly good job. I have listed below the pin to pin connection that are needed to wire up the MX-80 to the SOL parallel port. The driver in SOLOS runs the MX-80 with no problems.

SOL DB-25 Parallel port pins

1	2	16	17	18	19	20	21	22	23	24	25
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
7	23	11	1	9	8	7	6	5	4	3	2

MX-80 Amphenol 57-30360 connector

I recently aquired a Measurement Systems DMB 6400 memory board for a SOL. This is their bank select board and has a few features that the plain 6400 board does not have. It works perfectly in a SOL as a 48K board with 16K turned off. The best part is that you can cause a smaller amount of memory to be turned off by tricking the board to think that it's getting a not/phantom signal by making a ground connection. This allows the use of this board at 56K or 60K to go with the SOS module discussed above. At 48K it works perfectly with HELIOS.

Regards,

Gib Zeratsky

28 Bellevue Ave.
London, Ont., Canada
N6C 4A6

February 12, 1981

Dear Sirs:

I have recently added a Vista V-200 single drive disk system. Other than the CP/M manual, it has one of the worst documented items I have seen. Otherwise I am delighted. I use the single drive Filecopy routine by K. Barbier in September Kilabaud and along with the sector by sector COPY routine from Vista, this is a very reasonable single drive system for the SOL. The list price with all the CP/M 1.4 software is $695.00. I spent $619.00 plus shipping.

Some one should have said that the TAD Enterprises ECB-CP/M conversion permits the transfer of programs between ECB cassette files, ECB disk files, and CP/M ED files. That may be one of the best features of that package. I much prefer the ECB editor, and I can also do a lot of debugging with the interpreter before compiling with CBASIC2.

I have a copy of AIR TRAFFIC CONTROLER from Creative Computing. As a private pilot I think it is a realistic and super game, or more correctly, simulation! It works!! No bugs! Fine value at $8.00.

G2 Mocrosoft Basic would be a super interpreter for the SOL if it could write basic files in ECB T format, and if it could write string data files without integet translation. It is almost useless without these. Perhaps a USR subroutine with the ECB source for T files?

Has anyuone tried to use the SOL parallel port as the lister port? I have not been able to write a driver that does not hang the system on a control-P.

Yours truly,

J.D. Barber

Stan Sokolow
Editor
PROTEUS News
1690 Woodside Rd., Suite 219
Redwood City, CA
U.S.A. 94061

16 Marwin Rd.
Pickering, Ontario
Canada
L1V 2N7
31 January 1981

Dear Stan,

I am finally getting around to writing. It was your request for information on floppy discs compatible with the SOL in the Nov./Dec. 1980 issue that finally got me going. I have a SOL-20 with 40 K RAM consisting of 2 - Processor Tech. 16 K dynamic RAM boards and a 8 K Processor Tech. static RAM board. They function without problems. I bought a used Micropolis Mod II "quad density" floppy disc with dual drives about six months ago. It fits your description of a system that just plugs in without modifications. The controller board plugs into the backplane board, you EX F400, the bootleg loader loads the program, you do a keyboard reset, EN 81 HEX at a given location, EX another address and the system is configured. You then merely use the disc operating system to save the refigured DOS and you are off to the races. There was an excellent article on MDOS, the operating system in Kilobaud about a year ago and there is another fine article (written humbly by me!) that will be in an upcoming edition of Elementary Electronics that deals more with the hardware and the Micropolis BASIC. I find the system super from both a hardware and software point of view. My only early misgiving was that the address of the controller board is at F400 HEX which conflicts with ALS-8 but the Micropolis system comes with a disc based assembler that is only a little less useful than the ALS-8. I also have CP/M 1.4 for the system which configured the same way.

As a result of an article that I wrote in the June 1980 Kilobaud, I was swamped with SOL owners who were running all sorts of systems. I steered them all to PROTEUS and I hope they will write. North Star runs with SOL with not much help, Vista now configures a controller board ROM for SOL. I am aware of only one person using it but he seems quite happy. Another person has used Thinker Toys Discus II with the eight inch drives and seems happy. I wish these people would either write to you or even to one of the "regular" magazines such as Kilobaud. There must be hundreds of people hiding out there doing the most amazing things with their SOL's but keeping it to themselves.

A fellow in British Columbia just wrote to me that he has used the National Semiconductor NM58167 chip and a BSR X-10 with his SOL to control his house! I have been trying to do this using material from BYTE for about two years - where has this guy been hiding. I will encourage him to publish this information, even if I have to ghost write it for him.

In volume 3 #4, John Nickel wrote in about his "basket case" SOL. His first problem with the screen filling up with garbage when he signs on was mentioned in an earlier issue of PROTEUS. The earlier article suggested a number of chips that could be wrong. As I don't have a decent scope, I just replaced them and it ran like a charm - for two weeks. What I have found is that if I give the machine about five minutes to warm up, it clears which suggests an intermittent short circuit. I am too lazy to follow this up and deal with the problem merely by turning the SOL on five minutes earlier and fixing a cup of coffee. His third problem suggests a similar program on his IMSAI memory board. By the time, he has loaded TREK-80 and tried it and then loaded Diagnostic II, the machine has warmed up and the problem disappears. I would use the same solution.

Another problem that SOL users have is with overheating. There have been a number of suggestions made in earlier editions but I think my solution works quite well. I have preregulated the power supply on the backplane board. Again my lazy nature got the better of me and I just regulated the "+7V.+ almost anything line" which is the main culprit since it has to be terminally regulated on the board and is in much demand on static memory boards. I have written all the details in the January/February 1981 edition of Elementary Electronics.

I am presently trying to get one of the new Vista printers (as you can see from my typing, I need it!) and once I get it running I will drop you a note.

Yours truly,

Bruce Evans

BRE/mjf

Bruce Robert Evans, M.D.

Stanley M. Sokolow
Proteus News
1690 Woodside Rd., Suite 219
Redwood City, CA 94061

Dear Stan,

The following modification to the SOL Keyboard may be of help to other readers of Proteus. I had noticed that from time to time I would get double character entry which I thought was switch bounce. When I changed over to a Z-80 cpu board, (4MHz), the problem of double entry occured approximately 25% of the time. I checked out the strobe pulse from the SOL Keyboard and found it to last 11.5 msec. I changed R7 from a 1K ohm resistor to 680 ohms. This shortened the strobe pulse to 8 msec. which corrected the above problems. This change may help some of our members.

John R Gould

John R. Gould
720 Hadcock Rd.
Brunswick, Ohio 44212

---

429 HOLLYDALE CT., N.W.
ATLANTA, GA 3Ø342

JANUARY 25, 1981

PROTEUS NEWS
ATTN: STAN SOKOLOW
169Ø WOODSIDE ROAD, SUITE 219
REDWOOD CITY, CALIFORNIA 94Ø61

DEAR STAN:

AFTER QUIETLY FOLLOWING PROTEUS NEWS AND THE ADVICE OF FR. MCGAHEE FOR A LONG TIME I AM WRITING ABOUT A FEW ITEMS THAT I HAVE FOUND PARTICULARLY HELPFUL.

FIRST, ABOUT A YEAR AGO, I BOUGHT A VISTA V-2ØØ-2 DUAL DISK DRIVE SYSTEM BASED ON FR. MCGAHEE'S RECOMMENDATION IN AN EARLIER ARTICLE (KILOBAUD SEPT. '79). AS OTHERS HAVE ALREADY COMMENTED IN PROTEUS/NEWS MY EXPERIENCE WITH VISTA HAS ALSO BEEN VERY GOOD. THE SYSTEM WAS EASY TO INSTAL (NO MODIFICATIONS NEEDED) AND IT CAME WITH A VERSION OF CP/M TAILORED FOR THE SOL. NOW THAT OUR LIBRARY HAS CPMUG DISKS AVAILABLE ON CASSETTES I SEE NO REASON TO HESITATE IN RECOMMENDING THE VISTA DISK SYSTEM. ADD MY THANKS TO GREENLAW FOR MAKING HIS TAPEDISK AND DISKTAPE PROGRAMS AVAILABLE FOR PURCHASE. THE PEOPLE AT VISTA HAVE PRODUCED A QUALITY SYSTEM WITH THOROUGH DOCUMENTATION AT A REASONABLE PRICE.

ONE ADVANTAGE OF THE CP/M EDITOR THAT I HADN'T FORESEEN IS THE ABILITY TO COMPOSE LETTERS CENTERED ON THE 64 CHARACTER WIDE LINES OF THE VIDEO DISPLAY AND ADD 9 BLANKS (SPACES) TO THE BEGINNING OF EVERY LINE WITH A SINGLE MACRO COMMAND IN ED BEFORE COPYING THE LETTER ON MY 82 CHARACTER WIDE PRINTER. THE LETTERS COME OUT CENTERED ON THE PRINTED PAGE AND I HAVEN'T HAD TO MODIFY MY SOL.

FR. MCGAHEE HAD RECOMMENDED A MANUAL DESCRIBING BASIC-E. LAST YEAR I OBTAINED A COPY FOR $25.ØØ + 1.5Ø POSTAGE FROM THE JEM COMPANY, SUITE 3Ø1, 2555 LEAVENWORTH ST., SAN FRANCISCO, CA 94133. I FOUND THIS MANUAL TO BE VERY HELPFUL IN USING BASIC-E. ALSO, THERE IS AN ARTICLE IN THE JANUARY 1979 ISSUE OF BYTE THAT DESCRIBES THE USE OF DATA FILES IN BASIC-E. THE ARTICLE STARTS ON PAGE 84 AND IS TITLED "A COMPUTERIZED MAILING LIST". IT IS AN EXCELLENT SUPPLEMENT TO THE USER'S GUIDE FOR BASIC-E FROM THE JEM COMPANY.

SINCERELY,

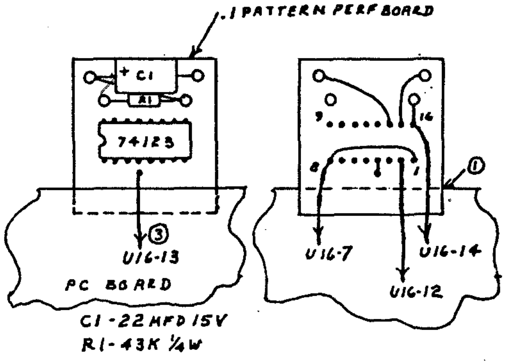
Al McSweeney

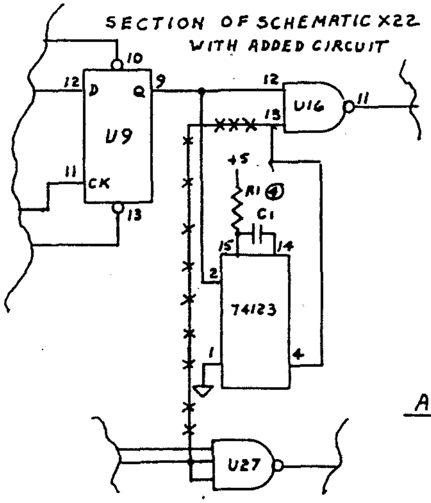
AL MCSWEENEY

AUTOMATIC REPEATING MODIFICATION FOR SOL KEYBOARD

by Jack Kinney

This is a one-IC piggyback for the Sol keyboard that makes any key repeat automatically if held down beyond a certain delay. R1 may be varied to select the desired delay. Suggested values are shown below.





NOTES:

① BOARD MAY BE GLUED TO BOTTOM OF PC BOARD DIRECTLY ABOVE U-16.

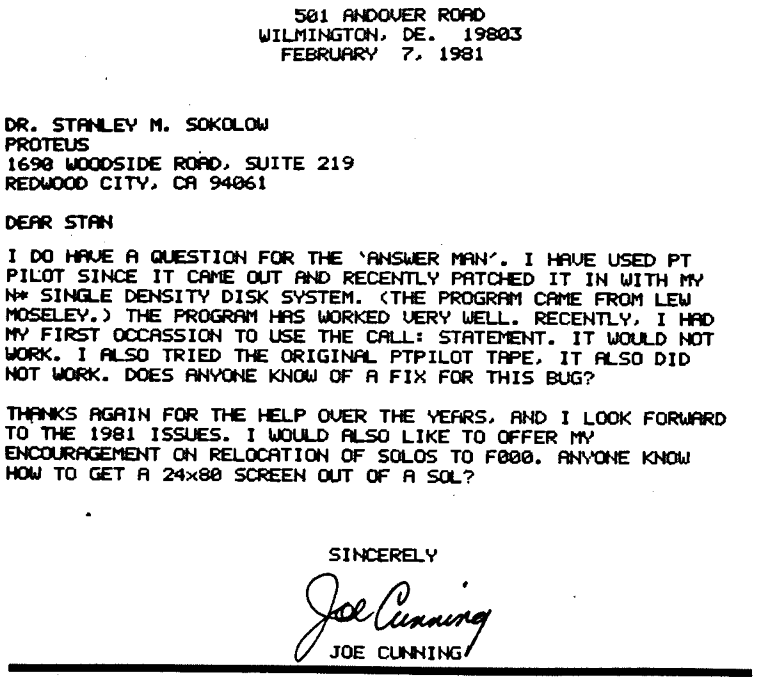
② ON LATER KEYBOARD WITHOUT SOCKETS, U16 MUST BE REMOVED AND A SOCKET INSTALLED.

③ PIN 13 OF U16 IS BENT OUT TO DISCONNECT IT FROM EXISTING TRACE. THE WIRE FROM PIN 4 OF THE 74123 IS SOLDERED DIRECTLY TO PIN 13. THIS AVOIDS TRACE CUTTING.

④ R1 MAY VARIED FOR DESIRED DELAY.

AN AUTOMATIC KEY REPEAT FOR SOL

JACK KINNEY
UCLA COMPUTER SCIENCE DEPT
3413 BOELTER HALL
LOS ANGELES CA 90024



501 ANDOVER ROAD
WILMINGTON, DE. 19803
FEBRUARY 7, 1981

DR. STANLEY M. SOKOLOW
PROTEUS
1690 WOODSIDE ROAD, SUITE 219
REDWOOD CITY, CA 94061

DEAR STAN

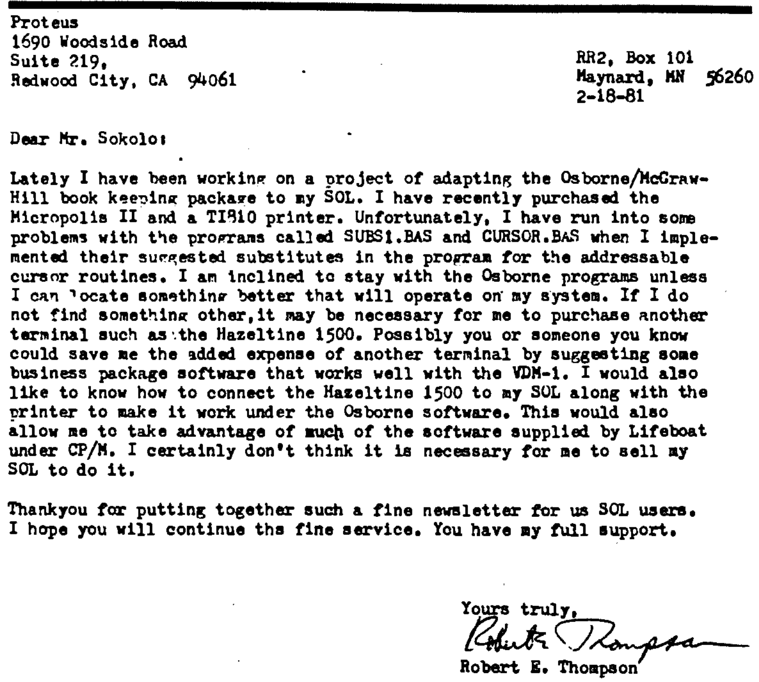
I DO HAVE A QUESTION FOR THE 'ANSWER MAN'. I HAVE USED PT PILOT SINCE IT CAME OUT AND RECENTLY PATCHED IT IN WITH MY N* SINGLE DENSITY DISK SYSTEM. (THE PROGRAM CAME FROM LEW MOSELEY.) THE PROGRAM HAS WORKED VERY WELL. RECENTLY, I HAD MY FIRST OCCASSION TO USE THE CALL: STATEMENT. IT WOULD NOT WORK. I ALSO TRIED THE ORIGINAL PTPILOT TAPE, IT ALSO DID NOT WORK. DOES ANYONE KNOW OF A FIX FOR THIS BUG?

THANKS AGAIN FOR THE HELP OVER THE YEARS, AND I LOOK FORWARD TO THE 1981 ISSUES. I WOULD ALSO LIKE TO OFFER MY ENCOURAGEMENT ON RELOCATION OF SOLOS TO F000. ANYONE KNOW HOW TO GET A 24x80 SCREEN OUT OF A SOL?

SINCERELY

Joe Cunning

JOE CUNNING



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

RR2, Box 101
Maynard, MN 56260
2-18-81

Dear Mr. Sokolo:

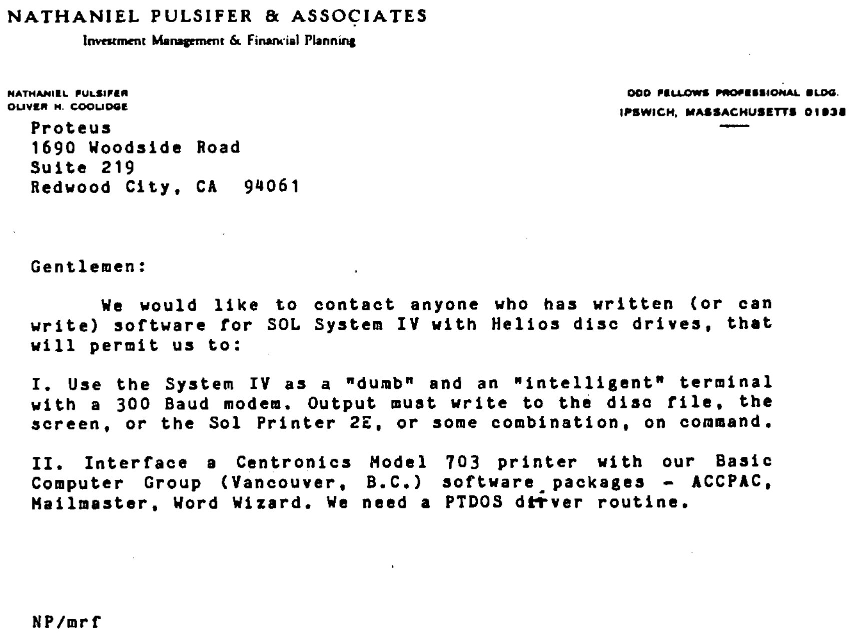
Lately I have been working on a project of adapting the Osborne/McGraw-Hill book keeping package to my SOL. I have recently purchased the Micropolis II and a TI810 printer. Unfortunately, I have run into some problems with the programs called SUBS1.BAS and CURSOR.BAS when I implemented their suggested substitutes in the program for the addressable cursor routines. I am inclined to stay with the Osborne programs unless I can locate something better that will operate on my system. If I do not find something other,it may be necessary for me to purchase another terminal such as the Hazeltine 1500. Possibly you or someone you know could save me the added expense of another terminal by suggesting some business package software that works well with the VDM-1. I would also like to know how to connect the Hazeltine 1500 to my SOL along with the printer to make it work under the Osborne software. This would also allow me to take advantage of much of the software supplied by Lifeboat under CP/M. I certainly don't think it is necessary for me to sell my SOL to do it.

Thankyou for putting together such a fine newsletter for us SOL users. I hope you will continue the fine service. You have my full support.

Yours truly,

Robert E. Thompson

Robert E. Thompson



NATHANIEL PULSIFER & ASSOCIATES
Investment Management & Financial Planning

NATHANIEL PULSIFER
OLIVER H. COOLIDGE

ODD FELLOWS PROFESSIONAL BLDG.
IPSWICH, MASSACHUSETTS 01938

Proteus
1690 Woodside Road
Suite 219
Redwood City, CA 94061

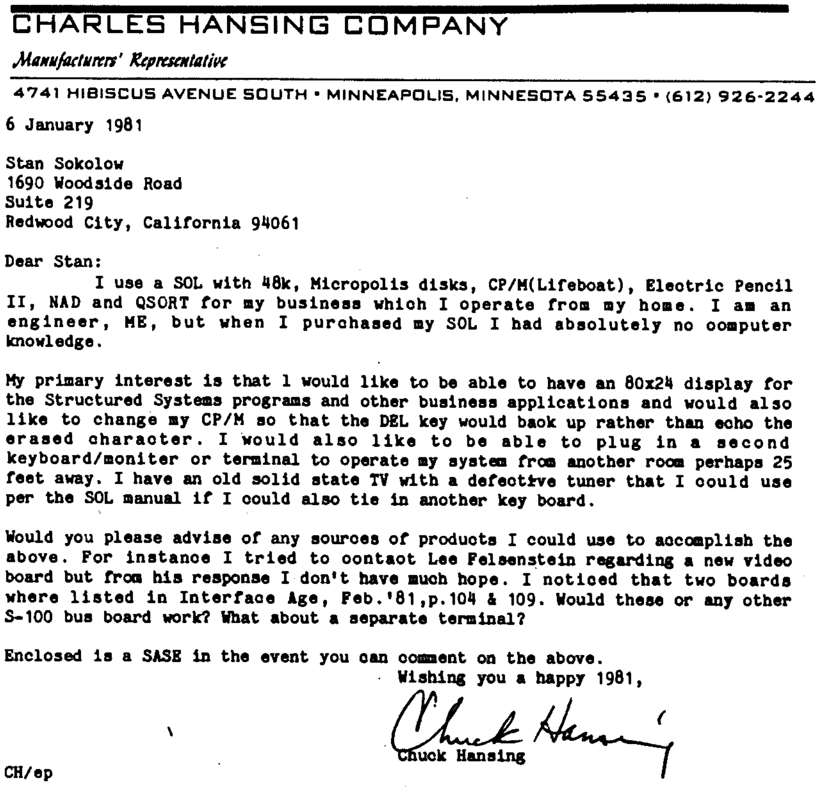
Gentlemen:

We would like to contact anyone who has written (or can write) software for SOL System IV with Helios disc drives, that will permit us to:

I. Use the System IV as a "dumb" and an "intelligent" terminal with a 300 Baud modem. Output must write to the disc file, the screen, or the Sol Printer 2E, or some combination, on command.

II. Interface a Centronics Model 703 printer with our Basic Computer Group (Vancouver, B.C.) software packages - ACCPAC, Mailmaster, Word Wizard. We need a PTDOS driver routine.

NP/mrf



CHARLES HANSING COMPANY
Manufacturers' Representative
4741 HIBISCUS AVENUE SOUTH • MINNEAPOLIS, MINNESOTA 55435 • (612) 926-2244

6 January 1981

Stan Sokolow
1690 Woodside Road
Suite 219
Redwood City, California 94061

Dear Stan:

I use a SOL with 48k, Micropolis disks, CP/M(Lifeboat), Electric Pencil II, NAD and QSORT for my business which I operate from my home. I am an engineer, ME, but when I purchased my SOL I had absolutely no computer knowledge.

My primary interest is that l would like to be able to have an 80x24 display for the Structured Systems programs and other business applications and would also like to change my CP/M so that the DEL key would back up rather than echo the erased character. I would also like to be able to plug in a second keyboard/moniter or terminal to operate my system from another room perhaps 25 feet away. I have an old solid state TV with a defective tuner that I could use per the SOL manual if I could also tie in another key board.

Would you please advise of any sources of products I could use to accomplish the above. For instance I tried to contact Lee Felsenstein regarding a new video board but from his response I don't have much hope. I noticed that two boards where listed in Interface Age, Feb.'81,p.104 & 109. Would these or any other S-100 bus board work? What about a separate terminal?

Enclosed is a SASE in the event you can comment on the above.

Wishing you a happy 1981,

Chuck Hansing

Chuck Hansing

CH/ep

December 14, 1980

Dear Stan:

It has now been one year since I have been a member of your organization so I don't know what has been happening or if you still exit. I have recently put a mod in my SOL that other users may be interested in so I am writing this letter. Since I have been using disk based assebmlers I have been constantly bothered about the wasted space in high core. Since CP/M requires contiguous memory we all have been limited to 48K systems. However:

Install a temporary jumper between U24-2 and U22-5,9. This will re-address the personality module, the video RAM and the scratchpad RAM into the F block. Re-assemble SOLOS to start at F000 and the RAM to start at F800. Then make the following patches based on the new addresses:

```
F000    7F       MAKE LIKE CUTER
F037    7F       MAKE LIKE CUTER
F0E0    00       ADDRESS ABOVE VIDEO RAM
F0E1    C2       USE JNZ HERE
```

And re-define VDMEM to FC00.

The source code is available on disk from the CP/M users group for $8.00 so you don't have to type in the entire source code if you are running CP/M.

With these changes software designed for CUTER which does not use the memory mapping capability of the SOL will work directly. Software written of SOLOS which does not require memory mapping video will have to be patched to the new jump table in the F block. Unfortunately considerable patching is required for all of PTC's delighful video games. I installed my jumper with a switch and have two personality modules so that when I want to play games I can head back down to the C block.

GOOD LUCK

BRUCE BARRON
227 E. Marconi Ave.
Phoenix, AZ 85022
(Editor's note: See the article in this issue on the relocation of the Sol's address space to F000.)

8022 - 117th Street
Delta, B. C. V4C 6A9
CANADA
February 2, 1981

Dear Stan:

Enclosed is the print driver that I wrote for my ASL2 Multiwriter II. The heart of this printer is the Diablo Hytype I. When I obtained this printer, it was being run at 300 baud on the serial port. The following driver allows virtually any baud rate to 9600 baud and certainly has improved the efficiency of the throughput since implemented. At any rate, here it is for anyone who may have this printer. If someone has a better idea, I'm open to suggestion.

Note: The printer has a 64 character FIFO buffer.
The routine is written as CP/M List routine, data being sent out is moved in register C, not B as in SOLOS.

```
LIST    PUSH    H
        CALL    SEROUT  ;SEND A CHARACTER OUT
        LDA     CHRCNT  ;GET CHARACTER COUNT
        INR     A       ;BUMP THE COUNT
        STA     CHRCNT  ;STORE NEW COUNT
        CPI     60      ;SENT 60 CHARACTERS?
        JZ      WAIT    ;IF SO STAND-BY
        CPI     55      ;IF <60, IS IT 55?
        JNZ     BACK    ;IF NOT, GET MORE
        MVI C,  5       ;IF=55, SEND OUT BUFFER MARKER
        CALL    SEROUT  ;TO HYTYPE I
BACK    POP     H

        RET             ;BACK TO MAINLINE FOR MORE STUFF
WAIT    IN      0F8H    ;GET STATUS
        ANI     40      ;MASK FOR SOL SERIAL DATA READY
        JZ      WAIT    ;LOOP AROUND IF NOT READY
        IN      0F9H    ;GET DATA
        ANI     07FH    ;LOP OFF THE HIGH ORDER BIT
        CPI     07FH    ;IS IT HYTYPE I ANS-BAK?
        JNZ     WAIT    ;IF NO, CHECK AGAIN
        XRA     A       ;AFTER ANS-BAK, RESET
        STA     CHRCNT  ;ZERO THE COUNTER
        JMP     BACK    ;RETURN FOR MORE
SEROUT  IN      0F8H    ;GET STATUS
        RAL             ;PUT HIGH BIT IN CARRY
        JNC     SEROUT  ;GO ROUND AGAIN
        MOV     A,C     ;GET CHARACTER READY
        OUT     0F9H    ;SEND IT OUT
        RET
;
CHRCNT  DB      00H
```

Keep up the good work.

Yours truly

Wayne Wilson

Wayne Wilson

PROPRIETARY SOFTWARE

Proteus item P1:

WordWizard electronic typing system for the Sol with Helios disk, by BSG. Latest version is 4.0.2 system disk. See description in Proteus News, Vol. 2, Number 6.

On Helios disk with manual	$300.00
Manual only	$35.00

Proteus item P2:

MailMaster list management system for Sol with Helios disk, by BSG. Latest version is 3.0.2 system. See description in Proteus News, Vol 2, Number 6.

On Helios disk with manual	$400.00
Manual only	$35.00

Proteus item P3:

MailSort supplement to MailMaster, by BSG. Latest version is 1.0.1 system. See description in Proteus News, Vol 2, Number 6.

On Helios disk	$150.00
Manual only	$25.00

Proteus item P4:

AccPac General Ledger and Financial Reporting system for Sol with Helios disk, by BSG. Latest version is 1.3.4 G/L system, 1.1.2 F/R system disk. See description in Proteus News, Vol 2, No 6.

On Helios disk:	$600.00
Manual only	$35.00

Proteus item P5:

Acc Pac Accounts/Receivable system for Sol with Helios disk, by BSG. Latest version is 1.0.1 Daily disk, 1.0.1 Monthly system disk. See description in Proteus News, Vol 2, No 6.

On Helios disk:	$600.00
Manual only	$35.00

Proteus item P6:

AccPac Accounts/Payable system for Sol with Helios disk, by BSG. Latest version is 1.0.2 Daily, 1.0.3 Monthly. See description in Proteus News, Vol 2, No 6.

On Helios disk:	$600.00
Manual only	$35.00

Proteus item P7:

AccPac Programmer's Package for creating custom AccPac programs, by BSG. Latest version is 1.1.0 system disk. Requires PTC Disk Fortran system. See description in Proteus News, Vol 2, No 6.

On Helios disk:	$150.00
Manual only	$25.00

Proteus item P8:

(Discontinued)

Proteus item P9:

(Discontinued)

PROCESSOR TECHNOLOGY SOFTWARE
SOURCE AND OBJECT CODE

Note: These items are made available under license from Processor Technology Corporation. The terms of Proteus's license require that we have users of the SOURCE CODE sign a specific agreement. A license fee is required for two reasons: to defray the cost of acquiring the source code and to generate revenue for Proteus operation, including compensation of people who do significant work on repairing, maintaining, and enhancing the software.

The personal use license allows the licensee to use the code, regardless of how it is obtained (by purchase from Proteus, copied from another licensee's disk, etc.). So two or more Proteus members may join together and share one set of media, but each user must be licensed.

A commercial use license is available at addition cost. It allows distribution of object code derived from the source, with no royalties to Proteus or Processor Tech. Request price quotation.

Proteus item P10.
Source code for cassette-to-memory assembler Software #1, cassette-to-cassette assembler ASSM, PACK and UNPACK tape reformatting routines. Object code for FOCAL interpreter (source is not available).

License for personal use (required to purchase software on media below)	$50.00
On Helios disk	$15.00
On Sol/CUTS cassette in CTAPE format (C-90)	$15.00
Assembly listings (approximately 267 pages)	$40.00
ASSM manual	$ 3.00
FOCAL manual (25 pages)	$ 3.75
SOFTWARE #1 manual	$ 3.00
PACK/UNPACK instruction sheets	$ .50

Proteus item P11.
Source code not available. Object code for GAMEPAC 1, GAMEPAC 2, and TREK-80.

On Helios disk, runs on Sol/Helios	$15.00
On Sol/CUTS cassette, runs with SOLOS/CUTER	$10.00
GAMEPAC 1 manual	$ 3.00
GAMEPAC 2 manual	$ 3.75
TREK-80 manual	$ 3.00

Proteus Item P12.
Source code for Optional Precision Disk BASIC 1.1 mod 0, Disk BASIC/5, and Extended Cassette BASIC without MAT (matrix equation) statement. (WE HOPE TO PUT MAT STATEMENT BACK INTO E.C.BASIC AND RELEASE THE UPDATE AS SOON AS WE CAN.)

License for personal use (required to purchase software on media below)	$100.00
On Helios disks (2)	$30.00
On standard CP/M disks (2)	$30.00
On cassette	RPQ
Extended Cassette BASIC manual	$20.00
Extended Disk BASIC manual	$20.00
Opt Precision BASIC addendum for E. D. BASIC manual	$ 1.00
Disk BASIC/5 manual	$ 5.00

"RPQ" means "request price quotation".

Proteus Item P13.
Source code for Extended Cassette BASIC only, without MATrix statement.

License for personal use (required to purchase software on media below)	$50.00
On (1) Helios disk	$15.00
On (1) C-90 Sol/CUTS cassette in CTAPE format	$15.00
Assembled listing (127 pages)	$20.00

Proteus Item P14.
Source code and object code for ALS-8 program development system consisting of a memory-to-memory assembler, video editor, and 8080 simulator.

License for personal use (required to purchase software on media below)	$50.00
On (1) Helios disk or standard CP/M disk	$15.00
On (1) C-90 cassette in CTAPE format	$15.00
On (1) C-90 cassette in Greenlaw's format for CP/M	$15.00
Source listing (about 130 pages)	$20.00
ALS-8 manual	$ 7.50

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

License for personal use (required to purchase software on media below)	$100.00
On (3) Helios disks or (4) standard CP/M disks	$45.00
On (10) Sol/CUTS C-60 cassettes in Archive format	$100.00
On Sol/CUTS C-60 cassettes in Greenlaw's format	RPQ
A sembly listing of RESIDENT file (200 pages)	$35.00
Assembly listing of GET command which reads Archive format cassettes and CTAPE tape driver routine. (These are necessary if you want to write a program to read Archive tapes into a non-PTDOS system.)	$ 9.00

Proteus Item P16.
Containing SIMUciser, ParaSol driver, DISKT Helios diagnostic test, DCHECK file checker for PTDOS, and memory tests.

License for personal use (required to purchase software on media below)	$25.00
On Helios disk	$15.00
On C-90 cassette	$15.00

Documentation of these programs will appear in the Proteus "Encyclopedia Processor Technica" to be published in 1981.

Pr teus Item P17.

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

On Sol/CUTS cassette in eXEcutable form, with manual	$30.00

Proteus Item P18.
ASSM cassette-to-cassette assembler that will assemble source files too large to fit into memory in one load. Works well in conjunction with EDIT above.

On Sol/CUTS cassette in eXEcutable form, with manual	$30.00

OTHER PROPRIETARY SOFTWARE

Proteus Item P19.
Nevada COBOL compiler, a subset of ANSI-74 COBOL. Two versions available: for the PTDOS disk operating system (frozen at version 1.8), or for CP/M disk operating system. By Ellis Computing.

on Helios disk for PTDOS	$99.95
on Helios disk for Lifeboat CP/M	$99.95
on CP/M 8" standard diskette	$99.95
on NorthStar single density for CP/M	$99.95

Proteus Item P20.
Disk PILOT, an extended version of PILOT computer dialog language. Designed for computer aided instruction, including control of video cassette recorder connected to Sol parallel port. Present versions require a Sol+Helios or Sol+NorthStar. Other versions under development by John Starkweather.

On Helios disk for PTDOS	$99.95
On Helios disk for Lifeboat CP/M	$99.95
On NorthStar disk for CP/M (specify density: SD/DD)	$99.95

SALES TAX REMINDER

When ordering items from Proteus, members in California should remember to add sales tax. Periodicals are not subject to sales tax, but computer hardware, software, books, and pamphlets are taxable if sold to a California address. Our update service, where we upgrade a program on your disk to a more recent revision level, is considered a non-taxable labor item. Subscriptions are not taxable.

SOL AND HELIOS SERVICE IN LOS ANGELES AREA

Jack Kinney and Steve Sakamoto have had extensive experience in servicing the Helios disk and Sol computers owned by UCLA Computer Science Department. They are offering their services to any Sol owners in their area.

Jack has written that UCLA had about 45 Sols in many departments since 1976, so they have really learned all about Sol service. When PTC was in business, Jack obtained all of their available technical aids developed for factory and dealer use.

Jack also designs custom interfaces, primarily for the experimental Psychology department at UCLA, and he is available on a limited basis for outside work on his own time.

If you need someone who knows what he's doing and has the equipment, contact Jack Kinney, Biotronic Designs, 5660 Topeka Drive, Tarzana, CA 91356, (213) 345-9775 (evenings and weekends).

Also, see the modification Jack has contributed to this issue of the newsletter. It provides an automatic repeat function that repeats any key after it is held down continuously for a certain period of time, much like the automatic repeating keys on a typewriter.

(Editor's note: Jack actually wrote to us long ago, but his letter got delayed in the shuffle between editors last year. Sorry, Jack, but we're back on track now.)

TABLE OF CONTENTS

FLASH! BUSINESS BASIC FOUND

When we distributed the source code license and order form recently, we omitted Business Basic bacause it wasn't among the disks we received. But through a stroke of luck, we have obtained a disk with Basic 1.2 mod 1, "Level I Business BASIC" source code just recently. Members who have purchased the license to use PTC Disk BASIC source code (Item P12 license) may add this Business BASIC disk to their order for the software on Helios or CP/M disk, at $15 per disk. This BASIC was never released by PTC except in a preliminary demo version, so it may need tender loving care. For that reason, it is not a substitute for the Optional Precision BASIC. We also have a write-up on Business BASIC Demo version for $2.

This BASIC provides the features of Optional Precision BASIC 1.1 plus CHAINing (similar to XEQ but preserves variables in COMMON area), COM statement to declare variables in Common area that is unchanged when chaining to another BASIC program module, "exact" input option on INPUT statements, SYSTEM statement that provides various options such as disable MODE SELECT key interruption of running programs, etc., the SYST function which allows testing of system parameters. It also allows error routines to return to the normal flow after handling the error, and it corrects a bug in user defined functions.

PROTEUS/NEWS

A news journal for owners and users of Processor Technology Corporation computer equipment. Published by Proteus, 1690 Woodside Road, Suite 219, Redwood City, California 94061, USA.

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

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

Joe Maguire
PO Box 3742 DT
Anchorage, AK

99510

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

March/April 1981 PUBLISHED BIMONTHLY BY PROTEUS, 1690 WOODSIDE ROAD, SUITE 219, REDWOOD CITY, CA 94061, USA Vol. 4, #2
Single Issue...$4 (US)

NEWS FROM THE COMPUTER FAIRE

by Stan Sokolow

Through the courtesy of the Computer Faire management, Proteus had a meeting room and exhibit booth in the 6th West Coast Computer Faire, April 3-6, in San Francisco, California. This year I had a much more pleasant experience at the Faire than in the past, because I didn't try to do too much. Many members came by to say Hello and ask questions. Numerous newcomers also came by, some who recently bought a Sol and others who just never knew we existed.

The Faire was larger than ever. Over 30,000 registrations were sold--more than they expected. The Faire ran out of programs on Saturday and had to find a printer open on weekends to do a rush job to accommodate the expected Sunday crowd. Pre-Faire radio coverage was excellent. One of the large retailers in the San Francisco area gave many plugs to the Faire in the two weeks before. I guess it paid off for them, because I saw two young men trucking in dozens of Apple computers and overheard that they sold 30 Apples on Saturday alone.

There were a few new computers on exhibit this year, but definitely less excitement was shown than in the early days of personal computing. The biggest interest was around the new Osborne portable system. This is a Z-80, 64K, system with two minifloppies, keyboard, and 5" video monitor, all in a rugged aluminum case that closes up to be carried like luggage. They plan to have a battery pack for it, which will fasten onto the side and allow it to operate 5 hours on one charging. The idea is that business people will be able to carry the computer on-board with them on airline flights, and use the system in the air. (FCC and CAB approval is being sought by Osborne.) The size meets CAB regulations for under-the-seat carry-on luggage. Another interesting point is that the system sells for less than $1800, quantity one, including CP/M and application software tools (a wordprocessor, Microsoft BASIC, a VisiCalc-like program for financial worksheets, etc.).

Another interesting machine was the Micro-Expander, which we've mentioned before in Proteus/News. This is a Sol-like configuration (without the walnut sides though) containing a Z-80, IEEE standard S-100 bus with 4 slots, real-time clock, serial & parallel ports, TRS-80 format cassette interface, 24x80 color video interface, integral keyboard, and an internal expansion bus allowing memory and peripheral expansion without using up S-100 bus slots. Including a 64K S-100 memory board, they were selling this computer for $1875 at the Faire (reg. $2200) including Microsoft BASIC.

It is a high price for a bare-bones computer, but when you put a disk and monitor on it, you have a cost-effective system that has more features than any other on the market near that price. They are talking about adding RGB color output, which allows vivid color display available only on high priced graphics equipment. The signals already are available on the computer; all they need is to add a connector.

The surprising thing about both of these new computers is that they are both products of the same man who designed the Sol! The same theme can be seen running through all 3 computers: the all-in-one package concept of terminal integrated into a quality computer system. Lee Felsenstein has been prolific, and we hope that he finally reaps some financial rewards commensurate with the quality of his work. The Apple boys (Wozniak and Jobs) made a clever little machine, but their Apple II can't compare to the computers done by Lee. Maybe this time Lee will strike the silicon gold-mine. (Wozniak and Jobs began Apple while in their late teens and early twenties. A few years later they were multi-millionaires when Apple became a publicly-owned corporation.)

For more information on these new computers, contact Micro-Expander, Inc., 6835 W. Higgins Ave., Chicago, ILL 60656, (312) 792-1196, and Osborne Computers Corporation, 26500 Corporate Ave., Hayward, CA 94545.

Another booth at the Faire caught my attention, and I can recommend these people since I have subsequently used their service. I have an Epson MX-80 printer and a Diablo daisy-wheel printer. The cost of keeping these printers fed with ribbons has been significant. The Epson ribbon cartridge lists for about $18 and Diablo-compatible ribbons go for $6 to $8. Finally, I found a company that will reload these cartridges with fresh ribbon, and they will handle small quantities. I had an MX-80 ribbon reloaded for only $4.50 and HyType II ribbons refilled for 2.60 to 3.12, depending upon the type of ribbon. (The brown carbon-film "multi-strike" ribbon looks great on my ivory letterhead. It was only $3.12.)

The company that did this is American Ink Products Company, 527 Howard Street, San Francisco, California 94105, (415) 982-0161. The prices I quoted may be slightly higher now, because there was a 10% discount at the Faire. If you ask for it, maybe they'll give you the discount on your first order. They have a complete price list showing all of the ribbons they process. Even if your printer isn't on the list, they may be able to quote a price. My order was filled promptly and the cartridges were all returned in sealed plastic bags to preserve freshness. I recommend them highly.

Another eye-catching booth was the Japanese computer exhibit. Strikingly bold color displays (RGB no doubt), well made consumer-electronics-type cases, and some familiar names, like "Sharp". They aren't ready to push into the American market, but they've already pushed American companies like Commodore out of the Japanese market for personal computers. When they are ready to jump in here, look out. Already they are strong competition in the printer market (Epson, C.Itoh). I hear Centronics is in trouble due to Japanese competition, and I don't doubt it. Epson is air-shipping literally tons of little printers into the US at a price that Centronics can't match.

INDUSTRY RUMORS

Datamation magazine (April 1981) reports that IBM has denied the rumors that it is working on a small computer venture with the Japanese firm of Matsushita, but the Datamation reporter claims IBM will introduce a desk-top computer this summer. It is aimed directly at the Apple, RadioShack, and Commodore computers. Rumored to have a price tag under $5000, it supposedly will be an Intel 8088 processor.

What is Dr. Ken (UCSD Pascal) Bowles up to lately? Datamation says he's started a company called TeleSoftware Inc. to develop the first commercial "Ada" compiler by early 1982. "Ada" is the language adopted by the Department of Defense for all new government projects. The DoD has invested $10 million in developing Ada and will require it to be the computer language used by all Defense contractors on government work. It is bound to be the industry standard in the coming years. Datamation says Bowles moved to Ada due to the difficulties in using Pascal for distributed processing. Pascal was strongly influential in the design of Ada language.

Datamation (same issue) also has an article on DBMS (data-base managment systems) for microcomputers, which gives a brief but favorable description of the Micro Data Base Systems, Inc., package called "MDBS", which we've announced in prior issues of this newsletter. However, Computer Decisions magazine (pages 48-52, April 1981) reports on a study by Andrew Sharp of Tab Products, Inc., benchmarking MDBS on an inventory application he wrote for in-house use at Tab.

Using an 8085-based CP/M system with Microsoft BASIC, it took 9 seconds for the system to locate a record in a 1300 item file, 23 seconds to insert a single record into that file, and 63 minutes to insert a file of 260 records into the 1300 item file. Using MDBS's estimates, an analogous insert into a 10,000 item file would take 100 hours, Sharp estimated. And this is using a fast hard-disk drive, not a floppy!

For comparison, we use MailMaster to maintain our Proteus/News subscriber list. In the worst case, it would take about 2 seconds for our Sol to locate a record in a 1300 item file, and I believe the time would not be much different for a 5000 item file. And this is using a floppy disk system.

With all of its versatility, the MDBS system plus the Microsoft BASIC just must have too much processing overhead for the task. Sharp and the article's author Hazelton believe that the 8-bit word is the culprit, because the IBM 370/148 is no more than 20 times faster than the 8085 but doesn't suffer this slowness in that proportion.

Before you sink $1000-$2000 into MDBS and its options, be sure your application needs that much versatility. It may make the system so slow that it is unusable.

Product Announcement

SOL/NORTHSTAR USER'S GUIDE

Remember how difficult it was to know where to begin when you were first studying your user's manuals? Sol owners with NorthStar disk drives are faced with the Sol manual, the Solos operating system manual, the NorthStar BASIC manual, the NorthStar DOS manual, and so on. It can be difficult to know how to begin, because you need to know what you are doing before you can figure it out.

The Lindsay Group, back in 1978, tried to guide the novice through the learning process with a 43 page booklet called the "Sol 20 Meets the North Star Disk Drive." This covers the basics of how to turn on the machine, how to talk with the disk operating system, how to use the North Star BASIC, how to use Solos commands, etc. There is a table of commands telling you the command name, which layer of the system uses the command (Solos, DOS, BASIC) and where in the guide you can find a description of what it means. Lots of other information is summarized in the guide, so you don't have to hop through the manuals to find a frequently used item.

I don't know if the guide really clears up the technicalities for the novice, but it may be worth the price for you. Frank de Coster, a Proteus member, has a bunch of these booklets which he will sell just to recover printing costs at $7.50 each including mailing. (They originally retailed for $19.95). If you have a NorthStar/Sol combination, it may even be worth the price to buy one now as a selling point for some future time when you may want to sell the system.

If you want one, send $7.50 to Frank de Coster, 315 Wayne Place, Oakland, CA 94606.

Product Announcement

INDUSTRY STANDARD TAPE DRIVES

If you want to obtain data from universities, government agencies, public-domain software libraries, and the like, on ANSI-standard IBM-compatible 9-track tapes, you should investigate the interface board manufactured by Pacific Office Systems, 918 Industrial Avenue, Palo Alto, CA 94303, (415) 493-7455.

The POS-100 NRZI Formatter/Controller costs $995 in single quantity, and will enable an S-100 bus 8080 or Z-80 CPU to read and write 9-track magnetic tapes using a "Pertec standard" tape transport. (Apparently, the Pertec transport was so successful, several other manufacturers used an identical interface specification for their drives as well.) Additional transports can be daisy-chained onto one controller. The interface is actually two boards; one for the S-100 bus and one for the tape transport, with a ribbon cable to connect the two.

The price does not include the tape transport, but we have seen refurbished ones being sold for about $1500.

A 2 MHz 8080 can operate a 12.5 ips tape drive, and a 4 MHz can go to 25 ips (inches per second). Other drive speeds up to 37.5 ips can be obtained by software modifications. Timing is in software, so the Sol may require a slightly different delay routine, due to its clock frequency being 2.045 MHz with the 8080A microprocessor. You may be able to adjust for this slight difference when setting the tape speed on the transport. (Pacific Office Supply has a Sol, so specify Sol-compatiblity when you order, and they should be able to check it out in their shop.)

Software routines supplied with the controller allow the computer to rewind, forward space record, back space record, forward space file, etc.. Status of the drive can be read by CPU, including off-line/on-line, invalid command, density, beginning of tape error, write-protect error, noise record, etc.. Approximately 2K bytes of RAM is needed for all of these routines.

A set of CP/M utilities for diskette to tape, tape to diskette, tape to printer, etc. can be purchased for an additional $100.

POS is a very small company, so Caveat Emptor. But we know the hardware is actually quite simple and full documentation is said to accompany the boards, including theory of operation, testing procedures, service manual, and schematics.

Quantity discounts are available.

Product Announcement

STARKWEATHER'S

DISK PILOT LANGUAGE

FOR MAN/MACHINE DIALOGS

Reviewed by Stan Sokolow

Way back in the early days of interactive computer systems on big computers, researchers began working on computer-aided-instruction (CAI) projects. It soon became evident that the computer programming languages were not well suited to writing the interactive programs needed by teachers for programming a dialog with their on-line students.

One of the pioneers in CAI, John Starkweather, PhD, developed a simple programming language that gave the instructional author just enough computing power and a simple language for making concise programs. The language was not intended as a general-purpose computing language, so it was weak in calculation but strong in character string processing features. Starkweather named the language "PILOT", for Programmed Inquiry, Learning or Teaching, and it has become the most widespread language for CAI.

While Processor Technology was a rising star in the microcomputer field, Dr. Starkweather developed an 8080 implementation of the PILOT language and customized it for the Sol computer. (Dr. Starkweather is one of our members.) Processor Technology Corporation actually distributed the cassette version of PILOT, and the disk version was under development when PTC went out of business.

But Starkweather didn't stop development, and he now has released the language on several different disk formats, taking advantage of the extra features possible with a disk. These versions can be purchased through Proteus, as our item number P20, for $99.95 on PTDOS/Helios disk, or on Lifeboat-CP/M compatible Helios disk, or on CP/M-compatible NorthStar disk. The Sol/CUTS cassette version (SOLOS/CUTER compatible) is still available with original manual, as Proteus item P21 for $49.95, but it lacks some of the new features.

A company in Washington state called Micropi ("micro programmed instruction", I think) offers a version of PILOT called "Common PILOT" for CP/M, NorthStar DOS, TRS-80, Helios II, and TERAK disks at various prices from $275 to $295. Considering that Starkweather's PILOT comes from the inventor, his price seems very reasonable in comparison.

Since few readers may know of the power of PILOT, here is a brief introduction.

The task that PILOT is really designed to do is this. An instructional program needs to let the author present some text or pictures, ask a question, accept a response from the student, analyse the response, and respond accordingly. If the students reply is equivalent to the correct answer expected or to some expected incorrect answers, the program should present instructional text (such as the correct answer, tutorial material, etc.) and branch off to an appropriate part of the instructional program.

What PILOT does for the author/programmer is that it eliminates the programming jargon and allows the author to concentrate on the sequence and content. PILOT also has some powerful string matching capabilities that let the computer decide if the student's response really is equivalent to one of the expected right or wrong answers. The program must be able to accept many variations of grammar, such as singular and plural, and recognize them as the right answer.

For example, if the answer you expect is "running", you may need to accept as equivalent the answers "run", "runs", and even some synonyms such as "jogging", or misspellings such as "joging". You also need to eliminate extraneous words and just look for the key word in the student's response, because you may get various forms of sentence fragments, such as "he is running", "he ran", "running, I think", "he would run", and so on.

To program all of these string manipulations in BASIC would take quite a few instructions and would really obscure the meaning of the program, which is to detect the right answer "running". In PILOT the "match" instruction does it all with this much work:

```
M: run, ran , jog
```

This instruction tells PILOT to examine the student's response (stored in memory) for any word beginning with "run", or the word "ran" not embedded in a longer word, or any word beginning with "jog". The entire response is scanned and the result of the match (yes or no) is set in an internal register for testing by subsequent instructions.

Based on the results of comparisons like these, the program can branch to other instructions, call subroutines, tally scores, record a log of the answers on tape or disk, present other data on the terminal, and so on. A limited amount of integer arithmetic can be done.

PILOT has had various dialects develop, just as BASIC has. A common core of instructions was standardized for PILOT. The core includes "T:" for type a string onto the terminal, "A:" for accept an answer, "M:" for match the pattern parameters against the accepted response, "J:" for jump (go to), "U:" for use (subroutine call), "E:" for end, "C:" for compute, "R:" for remark. All of the instructions can be made conditional by appending the letter "Y" or "N" which tests the result of the last recorded match done. For example, "JY:" means jump if the last match was successful, and simply "Y:" means type this if the last match was successful.

Strings in PILOT don't require quotes around them, since they are used so often. Instead, variables are identified with a leading character (such as $ for string variables and # for numeric ones) that distinguishes them from text in the string. For example,

```
*START
T: Please tell me your name.
A: $NAME
T: Hi, $NAME!
 : How old are you?
A: #a
T:Is it fun to be #a years old?
```

You can see that this is really a dialog programming language, and it can be used for any sort of dialog not just programmed instruction. The current fad of verbal games like "Adventure" could be programmed with some sophistication using PILOT, and the conciseness of the language would allow larger programs to fit into limited memory space. Even applications such as medical history-taking could be done this way.

Disk PILOT version 4 by Starkweather provides many extensions beyond the standard core instructions of PILOT. For example, with PILOT you can open and close files, read and write data files to record answers, control the cursor location on the screen, pause a measured number of seconds, query the system to discover free space available in memory and other system parameters, load and execute other PILOT programs, call machine language routines, and so on.

Starkweather has also extended the meaning of variable names, so that you can address variables indirectly. That is, a string variable can contain the name of the variable you actually want to use; such as "$$NAME", which means use the string whose variable name is in the variable called "$NAME". (This sort of indirection is not found in BASIC, but the MUMPS language has it. In future issues, we will talk more about MUMPS. See the MUMPS article elsewhere in this issue.)

There is even a version of Starkweather's PILOT which allows the Sol to control a particular model of Sony videocassette recorder to rapidly search for a desired point on the video and then play a certain number of frames of video! No special hardware interface is required, except a specially wired cable and plug to connect the remote control socket of the recorder to the Sol's parallel port. (If you want this version, be sure to specify when you order.)

PILOT contains its own Sol-video editor, similar to the one called "EDIT" in PTDOS or the one in ALS-8. Version 4.2 and 4.3 use a serial terminal rather than the display of the Sol or VDM module.

A PILOT library is operated by a PILOT users group. Write to us of you want to find out where you can get in touch with PILOT users.

MUMPS LA"GUAGE FOR CP/M

You've heard of UCSD Pascal, but have you heard about UCD MUMPS? (That's not a typographical error. UCD=University of California, Davis.)

MUMPS is an ANSI-standard language (along with FORTRAN, COBOL, PL/I) that is available on many mini-computers, notably the DEC PDP-11 line. There is a small, fervent, and growing following of MUMPS users who feel that MUMPS is the greatest thing since sliced bread. Some have compared the time required to develop major programming projects, such as a complete hospital information system, with MUMPS versus with COBOL or FORTRAN, and MUMPS has been the winner.

Well, for the past few years, a small crew at University of California, Davis, has been working on an 8080 implementation of MUMPS, and they have distributed about 200 copies to date. The Department of Community Health at UCD is using the language for medical records and education. A number of public domain applications are being adapted to work on the system.

We have just received a letter announcing that the University will provide for an annual rate of $93 the latest copy of 8080 Standard MUMPS for CP/M on 8" diskette, along with the documentation manuals, and three times per year will send you an updated version of the object code with new or revised applications. Source code is available for $25 extra per copy. The letter is reproduced here.

There is an international MUMPS users group, called "MUG", which publishes a quarterly journal, holds annual scientific meetings around the US, publishes educational material on MUMPS, etc. Write to MUMPS User's Group, ℅ The Mitre Corporation, P.O. Box 208, Bedford, Mass. 01730.

HARD DISK FOR HELIOS

At the Computer Faire, I ran into two former employees of Processor Technology Corporation who mentioned that they are in the process of forming a company to service and upgrade Helios disk units. They plan to put an 8" Winchester hard disk drive into the empty side of the Helios II two-slot cabinet and use a Morrow Designs hard disk controller. Modification would be made to PTDOS to accommodate the new disk.

I mentioned that Proteus has the source code to PTDOS, and they were interested in working together with us. I know that many users are interested in adding the extra capacity of a hard disk, and this seems to be the ideal way if you already have a Helios. Those who do not have Helios would be able to use the regular 8" Morrow hard disk, which is available in 10 megabyte and soon in larger capacities.

When the company is ready, you will hear about it in this newsletter. One of PTC's best Helios technicians is involved in the company, so we know the work will be of expert quality.

24x80 VIDEO MODIFICATION FOR SOL

We have been told by a southern California manufacturer that a piggy-back board is being designed for modifying the Sol's video display to a standard 24x80 format. This is a modification, not a new display board which would require another slot in the bus. When it is available, Proteus will test it on our system and review the product in the newsletter.

The biggest handicap the Sol suffers right now is the small screen, since most pre-programmed application software assumes an 80 column screen. This modification board will bring Sol up to date. We're looking forward to it.

FORMER PTC ENGINEERS FORM SERVICE CENTER

A group of former Processor Technology technicians and engineers have formed a company for repairing, servicing, and customizing all of PTC's product line and most S-100 bus products. They are experienced in PTC hardware and software, including the Sol, Helios disk, PTC memory boards, printer interfaces, etc. They also have factory experience in Morrow and Dynabyte products, plus other experience in NorthStar, Godbout, Shugart, PerSci, Remex, Micropolis, etc.

They will service hardware, help with interfacing peripherals to computers, customize CP/M for specific hardware configurations, perform preventive maintenance, and so on. Helios alignment has a flat rate of $55 plus shipping; Helios repair flat rate $80 plus parts & shipping (unless the faulty part is not a field-replaceable item). Other service @ $40 per hour. They will give estimate before performing work, if requested.

Contact ACE Computers, Inc., 3388 Moraga Blvd, Lafayette, California 94549. (415) 283-6630.

(Editor's note: I know that when PTC had a tough one to repair, one of these fellows was the man whom they called upon. One of PTC's former corporate officers gave me nothing but the highest personal recommendation for his capability. So I think we are fortunate that this group has gone into the service business. All of these men have worked for some of the biggest names in S-100 microcomputing. -- Stan.)

PROTEUS CASSETTE SOFTWARE LIBRARY SERVICES

At the present time, the Proteus Cassette Software Library is offering services to Proteus members in 4 different areas:

First, we have the original Proteus Library cassettes. We now have available cassettes C1 - C11 (except C9, which has been delayed). These remain priced at $18 without a contribution, and $8 with an acceptable program contribution. These have been described in past issues of PROTEUS NEWS, or you may send me a SASE for a catalog sheet.

Next, we have on tape the full CPM Users Group library. At the present time, CPMUG disks 1 - 47 are available, with several more expected soon. A master catalog of these CPMUG volumes is available, either as hard copy or on tape, for $6.00. Unless you specify tape, you will be sent the hard copy. The CPMUG volumes are priced at $10.00 each (with or without a contribution).

In addition, there are another 7 disks from a new CP/M-oriented users group known as SIG/M. THis includes the famous ADVENTURE game, a 6502 simulator, and a number of CP/M utilities. The SIG/M disk catalog is included on the CPMUG catalog tape, and the SIG/M volumes are also priced at $10.00 each.

I expect soon to have additional volumes from the C Users' Group and one of the Pascal Users' Groups. If you are interested in these, you may write me.

The CPMUG and SIG/M software catalogs are provided on standard CUTS-format cassette tapes, and come with a special tape-to-disk loader program. To load the tapes, you need a SOL (or a compatable machine with a CUTS board and CUTER), some sort of disk system with CP/M v1.4 or v2.2, and your cassette recorder. You should have the motor control cable, too. Hard copy doc with the programs tell you how to get started.

Finally, we offer a CP/M file transfer service. If you have programs on a standard 8" single density CP/M disk, but have a Helios/CPM or a 5" disk system, we can transfer the programs from your disk to a tape format which you can then load into your disk system. The price for this service is $10.00 per 8" disk, regardless of how much or how little is on the disk. The price includes the cassette and postage, and the special tape-to-disk loader program. If you want your 8" disk returned, pack it in a reusable shipping box and enclose extra return postage (say, $1.00 for each 2 disks).

Send orders, inquiries and program contributions to:

PROTEUS Cassette Software Library
C/O Lewis Moseley, Jr., Librarian
2576 Glendale Court, NE.
Conyers, GA 30208

If you want a personal reply, please enclose a self-addressed stamped envelope. Enjoy!!!

PROTEUS LIBRARY CASSETTE C8: MORE ECBASIC PROGRAMS

THIS TAPE CONTAINS A NUMBER OF INTERESTING VIDEO GRAPHICS DEMONSTRATION PROGRAMS, AND ALSO SEVERAL SERIOUS PROGRAMS. EACH IS RECORDED TWICE ON SIDE 1 IN ECBASIC INTERNAL COMPILED FORMAT, AND ONCE ON SIDE 2 IN TEXT (PROTEUS STANDARD BYTE ACCESS) FORMAT.

#	NAME	TYPE	SIZE	DESCRIPTION
1	TCOPY	U	1K	OBJECT CODE FOR THE CUTS TAPE COPY/VERIFY PROGRAM BY LEWIS MOSELEY, JR. BRIEF INSTRUCTIONS FOR USE ARE INCLUDED. SOURCE IS ON CASSETTES C9 AND C10.
2	RNDCH	C	1K	RANDOM CHARACTERS ON THE SCREEN
3	RNDBW	C	1K	RANDOM BLACK AND WHITE ON SCREEN
4	SETSP	C	2K	OBSERVE EFFECTS OF 'SET DS='
5	DICE	C	3K	VIDEO DICE ROLLS ON THE SCREEN
6	BOGGL	C	2K	A NICE VIDEO IMPLEMENTATION OF THE BOARD GAME. ON-SCREEN TIMER. RULES NOT PROVIDED.
7	BINGO	C	2K	CALLS THE NUMBERS FOR A BINGO GAME AND RECORDS THEM ON THE SCREEN FOR CHECKING WINS.
8	DOODL	C	2K	DOODLE ON THE SCREEN. SELECT A CHARACTER AND LEAVE A TRAIL OF THEM ON THE SCREEN.
9	DATA	C	4K	A SOPHISTICATED REAL-TIME DATA GATHERING PROGRAM, CURRENTLY SET UP FOR MEDICAL MONITORING, BUT A GOOD EXAMPLE OF THE TECHNIQUE.
10	TTYPE	C	5K	TOUCH TYPING PRACTICE (WE ALL NEED IT!), WITH DIFFERENT LEVELS OF DIFFICULTY. FULL KEYBOARD SHOWN ON SCREEN.
11	CIRCL	C	1K	THE NEXT GROUP OF PROGRAMS, THROUGH LISJ9, ARE VIDEO GRAPHICS, AND SHOW THE PLOTTING OF A NUMBER OF TRIG FUNCTIONS. OBVIOUSLY, THE LISJ(N) PROGRAMS PLOT VARIOUS LISSAJOUS FUNCTIONS.
12	SPIRL	C	1K	
13	LISJ1	C	1K	
14	LISJ2	C	1K	
15	LISJ3	C	1K	
16	LISJ4	C	1K	
17	LISJ5	C	1K	
18	LISJ6	C	1K	
19	LISJ7	C	1K	
20	LISJ8	C	1K	
21	LISJ9	C	1K	
22	SEAWR	C	4K	GUNNERY PRACTICE AT SEA. CHOOSE THE RIGHT ELEVATION TO SINK THE ENEMY IN FLAMES.
23	FROG	C	3K	GRAPHICS FROG RACE, WITH BETTING, FOR MULTIPLE PLAYERS.
24	END	X	0K	TERMINATOR FOR TCOPY PROGRAM

USE THE TCOPY PROGRAM TO BACK UP YOUR CASSETTE.

PROTEUS LIBRARY CASSETTE C10: CP/M PROGRAMS FOR SOL

THIS IS OUR FIRST TAPE OF CP/M PROGRAMS, IN RECOGNITION OF THE FACT THAT MANY OF OUR MEMBERS HAVE GONE TO DISK. THERE ARE 5 MAJOR PROGRAM SYSTEMS, ANY ONE OF WHICH IS WORTH THE PRICE OF THE CASSETTE, AND SEVERAL UTILITIES. THE FIRST FILE ON THE TAPE IS THE 'TAPEDISK.COM' FILE, WHICH IS MANUALLY LOADED, AND WHICH THEN LOADS THE REST OF THE FILES TO THE DISK. ALL OF THE FILES HAVE THE SAME TAPE NAME - CPM - BUT DON'T WORRY; TAPEDISK KNOWS THEIR TRUE NAME. REFER TO THE HARD COPY DOCUMENTATION FOR LOADING INSTRUCTIONS.

```
 #   NAME      TYPE  SIZE   DESCRIPTION
---  --------  ----  ----   -----------
 1   BYTE      .ASM   5K    THIS ROUTINE, BY LEWIS MOSELEY,
 2   BYTE      .DOC   4K    ALLOWS CP/M'S PIP.COM TO READ A
                            PROTEUS STANDARD BYTE ACCESS TAPE
                            FILE ONTO THE CP/M DISK, OR VICE
                            VERSA.  NOW YOU CAN USE ALL OF THOSE
                            PROGRAMS YOU WROTE FOR ECBASIC.
                            ALSO, LOAD SOURCE FILES PROCESSED BY
                            'UNPACK', AND TRANSFER TO/FROM OTHER
                            SYSTEMS.
 3   COMLINK   .COM   6K    THIS PROGRAM INTERFACES CP/M WITH
 4   COMLINK   .DOC  30K    A D.C. HAYES 80-103A OR MICROMODEM
                            100 MODEM BOARD.  IT ALLOWS SOFTWARE
                            SELECTION OF LINE CHARACTERISTICS,
                            SUPPORTS AUTO DIAL AND AUTO ANSWER,
                            AND ALLOWS A TEXT FILE TO BE SENT TO
                            OR CAPTURED FROM THE MODEM.  FULL
                            SOURCE CODE IS ON TAPE C11.
 5   DISKTAPE  .ASM   5K    THIS GROUP OF PROGRAMS, BY GREENLAW,
 6   DISKTAPE  .COM   1K    ALLOWS THE TRANSFER OF ANY KIND OF
 7   DISKTAPE  .DOC  14K    CP/M DISK FILE BETWEEN DIFFERENT
 8   TAPEDISK  .ASM   7K    DISK SYSTEMS.  THE CP/M FILE IS
 9   TAPEDISK  .COM   2K    BROKEN INTO BLOCKS AND RECORDED TO
                            THE CASSETTE. VERY LARGE FILES CAN
                            BE TRANSFERRED, EVEN WITH A MINIMUM
                            16K SYSTEM. FULL INTERNAL ERROR
                            CHECKING. ALSO, A CHEAP, IF SLOW,
                            BACKUP FOR DISKS.
10   MFT       .ASM  17K    A VERY NICE SINGLE-DRIVE COPY
11   MFT       .COM   2K    PROGRAM BY LARRY HUGHES.  ALLOWS
12   MFT       .DOC   5K    COPYING ALL OR SPECIFIED PARTS OF A
                            DISK TO ANOTHER. USES ALL AVAILABLE
                            MEMORY AS A DISK BUFFER TO MINIMIZE
                            DISK SWAPPING.
13   TCOPY     .ASM  10K    CUTS TAPE COPY/VERIFY UTILITY BY
14   TCOPY     .COM   1K    LEWIS MOSELEY, JR., MODIFIED TO LOAD
                            FROM CP/M.  ALLOWS UNATTENDED
                            COPYING AND VERIFYING OF CASSETTE
                            TAPES. USE TO BACK UP CASSETTES.
                            OBJECT WILL BE INCLUDED ON MOST
                            FUTURE LIBRARY CASSETTES. DOCUMENTED
                            IN THE .ASM FILE.
15   PRINT     .COM   2K    A PRETTY-PRINTER WHICH GIVES NICE
16   PRINT     .DOC   5K    FORMATTED AND PAGINATED HARD COPY
                            FROM A DISK TEXT FILE. THE .DOC FILE
                            TELLS HOW TO CUSTOMIZE THE FORMAT TO
                            YOUR NEEDS.
17   WD        .COM   1K    TWO NICE EXTENDED DIRECTORY ROUTINES
18   XD        .COM   2K    TO SUPPLEMENT THE BUILT IN DIR
                            FUNCTION.  WDIR GIVES A 4 ACROSS
                            ALPHA-SORTED DIRECTORY.  XDIR GIVES
                            A COLUMNAR FORMAT, WITH FILE SIZES,
                            WHICH IS BEST FOR HARD COPY.  FROM
                            CPMUG.
19   PACK      .COM   2K    DONATED BY PROCESSOR TECHNOLOGY
20   UNPACK    .COM   2K    CORP., AND MODIFIED TO LOAD FROM
21   PACK      .DOC   3K    CP/M DISK.  THESE PROGRAMS CONVERT
                            BLOCK ACCESS FILES TO BYTE ACCESS
                            FILES, AND VICE VERSA.  THEY WERE
                            INCLUDED WITH PT'S EDIT AND CASSM
                            PROGRAMS.  USE TO CONVERT EXISTING
                            FILES FOR TRANSFER.
22   C10       .PRN   5K    THIS CATALOG LISTING
```

IN ALL, ABOUT 125K OF MATERIAL. ALL PROGRAMS ARE KNOWN TO WORK ON AN 8" SINGLE DENSITY SOFT SECTOR CP/M SYSTEM (TARBELL). ALL, WITH THE EXCEPTION OF XD.COM, ARE BELIEVED TO WORK ON ALL CP/M SYSTEMS. XD WILL PROBABLY GIVE SPURIOUS SIZE INFORMATION ON SYSTEMS OTHER THAN 8" SINGLE DENSITY SOFT SECTOR.

PROTEUS LIBRARY CASSETTE C11: MORE CP/M PROGRAMS FOR SOL

THIS TAPE IS PRIMARILY FOR PERSONS WHO WANT THE FULL SOURCE LISTING FOR COMLINK. THE SOURCE, ITSELF ALMOST 60K, WOULD NOT FIT ON C10. ALSO INCLUDED ARE THE TAPEDISK/DISKTAPE PROGRAMS TO LOAD THE TAPE TO CP/M DISK, AND A MEMORY TEST.

```
 #   NAME      TYPE  SIZE   DESCRIPTION
---  --------  ----  ----   -----------
 1   COMLINK   .ASM  57K    THIS PROGRAM INTERFACES CP/M WITH
 2   COMLINK   .COM   6K    A D.C. HAYES 80-103A OR MICROMODEM
 3   COMLINK   .DOC  30K    100 MODEM BOARD. IT ALLOWS SOFTWARE
                            SELECTION OF LINE CHARACTERISTICS,
                            SUPPORTS AUTO DIAL AND AUTO ANSWER,
                            AND ALLOWS A TEXT FILE TO BE SENT TO
                            OR CAPTURED FROM THE MODEM.
 4   DISKTAPE  .COM   1K    THESE PROGRAMS, BY RICHARD GREENLAW,
 5   TAPEDISK  .COM   2K    ALLOW THE TRANSFER OF ANY KIND OF
 6   DISKTAPE  .DOC  14K    CP/M FILE BETWEEN DIFFERENT DISK
                            SYSTEMS. THE CP/M FILE IS BROKEN
                            INTO BLOCKS AND RECORDED TO CASSETTE
                            BY DISKTAPE. TAPEDISK REVERSES THE
                            PROCESS. FULL SOURCE CODE ON
                            CASSETTE C10.
 7   QUIKTEST  .COM   1K    A MEMORY TEST PROGRAM WHICH TESTS
                            ALL MEMORY FROM THE START OF THE TPA
                            UP TO THE BASE OF THE CBIOS. RESET
                            AND REBOOT TO STOP THE TEST.
 8   TCOPY     .COM   1K    CUTS TAPE COPY/VERIFY UTILITY BY
                            LEWIS MOSELEY, JR. USE TO VERIFY
                            TAPE FILES AND TO MAKE BACKUP
                            COPIES. FULL SOURCE CODE ON C10.
 9   C11       .PRN   3K    THIS CATALOG LISTING
```

ALL PROGRAMS ARE KNOWN TO WORK ON AN 8" SINGLE DENSITY SOFT SECTOR CP/M SYSTEM, NORTHSTAR CP/M AND MICROPOLIS CP/M, AND ARE BELIEVED TO WORK ON ALL CP/M SYSTEMS.

CONTENTS OF HELIOS LIBRARY H5

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

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

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

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

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

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

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

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

CDC is driver for Control Data 9317 matrix printer.

SPIN is for NEC Spinwriter.

TI810 is for Texas Instruments 810 printer.

```
02/17/81  FILES ON: H5

  NAME       TYPE  SIZE   BLKZ     ID     SEC  TRK  ATTRI    INDEX
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !           IS     11    0100    0041     5   76
 1610        IW      4    04C0    0044     0    9   KWN
 CDC         IW      4    04C0    0046     8   14   KWN
 CONTENTS    .      12    04C0    0049     0   15
 CTAPE.A     T      64    04C0    001F     9    9   KWN
 CTAPE1      D       4    0100    001E     5    9   KN
 CTAPE2      D       4    0100    0020    13    9   KN
 DEC         IW      4    04C0    0047    12   14   KWN
 DEC.A       S      68    04C0    002E     0   37
 FEEDBACK    .       8    04C0    004A    12   15
 NOTICES     T       8    04C0    004C     8   16
 OKIDATA     T      16    04C0    003C     3    0   KWN
 SETPRT.A    T      20    04C0    001C     8    7
 SPIN        D       5    0100    003F     7    0
 SPIN:D      T      36    04C0    003D     0   18   K
 SPINWR:S    T      84    04C0    003E     4   20   K
 TERM:S      T       4    04C0    0028    12   22
 TERMINAL    I.      1    0100    002B    14   26
 TI810.A     T      36    04C0    0016    10    0
 WARRANTY    T       4    04C0    004B     0   16
 WPXER.A     S      44    04C0    002D     4   34
 WPXER.A2    S      60    04C0    002C     8   30
 XEROX.A     S      36    04C0    0029     0   23
 XEROX.A2    S      56    04C0    00.F     4   41
 m1610:s     S      36    04C0    0030    12   44
 m1610e:s    S      56    04C0    0031     0   47
 mHytype2    I.      3    0100    001A    14    0
 mS154C      I.      3    0100    0024    13   13
 mS154C:S    .      64    04C0    002A     6   26
 mSol2:S     .       4    04C0    0037     8   65
 mSol2E:S    .       4    04C0    0019    12    6
 mSol3:S     .       4    04C0    0038    12   65
 mSolp:S     .      24    04C0    0017     4    3
 msp2d:s     .      32    04C0    0018    12    4
 msp3d:s     .      12    04C0    0039     0   66
 w1610:s     S      44    04C0    0033     4   54
 w1610e:s    S      60    04C0    0032     8   50
 wSOL2       IW      8    04C0    0045     0   14
 wSOL2E      IW      8    04C0    0043     4    7
 wSOL3       IW      4    04C0    0048     0    7
 wSol2:S     .      40    04C0    0035     8   59
 wSol2E:S    .      40    04C0    003A    12   66
 wSol3:S     .      40    04C0    0034     0   57
 wsp2d:s     .      56    04C0    0036     0   62
 wsp2ed:s    .      44    04C0    003B     4   69
```

If you modify these programs and reassemble them, please remember that PTDOS requires all drivers to be loadable as one logical block. The assembler doesn't usually create image files in this format. To convert the object file into a single-block image file, use the command:
 EXTRACT file,S
which will "scrunch" the file into one block if possible. The scrunched file can be RETYPE'd into type "D" for driver. The physical blocksize doesn't matter, but you should chose a blocksize for efficiency. See the PTDOS manual for more info on this matter.

CONTENTS OF HELIOS LIBRARY H7

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

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

For further information please read the files WARRANTY and FEEDBACK.

Charles L. Athey, III
Proteus Librarian

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

ACCOUNTS - SAMPLE LIST OF ACCOUNTS

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

CREATFIL - CREATE RANDOM FILE OF ACCOUNTS

EXPENSES - SAMPLE LIST OF EXPENDITURES

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

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

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

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

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

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

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

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

MES.S is the source code for MESSAGE.

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

MES.TEMP is the source file for a message.

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

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

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

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

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

QUIKSORT same as above but with modifications suggested by Knuth.

HEAPSORT from Knuth

SHELSORT from Knuth

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

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

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

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

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

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

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

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

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

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

ROBOTS is an old game involving hiding from killer robots.

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

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

CPM-TXT Documentation in file CPM-TX.D

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

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

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

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

```
03/25/81  FILES ON: H-7

 NAME      TYPE  SIZE   BLKZ    ID     SEC  TRK  ATTRI    INDEX
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 Imessage   M     17    0240   0080    13   11  KWN       0B0F
 ACCOUNTS   08     5    04C0   004D     0    2  KWN       000F
 ADD.WK     05    12    04C0   009A     0   27  KWN
 ANYMO      08    33    0100   009B    13   20  KWN       1D07
 BANKERS    06    12    04C0   0050    12    3  KWN
 CONTENTS   P     28    04C0   0024    12   24  KWANEU
 CONVERT    05    36    04C0   009D     0   30  KWN
 CPM-TX.D   T      8    04C0   0097     9   20  KWN
 CPM-TXT    T      4    04C0   0096     5   20  KWN
 CREATFIL   06     4    04C0   004F     8    3  KWN
 EXPENSES   08    21    04C0   004E     4    2  KWN       010C
 FACTOR     05     4    04C0   0075    12    5  KWN
 FEEDBACK   T      8    04C0   0047     3    0  KWANEU
 FIND       05     8    04C0   0076     8    6  KWN
 FIND+      05     8    04C0   0077     0    7  KWN
 FORMAT     I.     8    09C0   0074     0    6  KWN
 FORMAT.S   T     40    04C0   007C     8    8  KWN
 HEAPSORT   05     4    04C0   0085    10   13  KWN
 KWIKSORT   05     4    04C0   0086     0   14  KWN
 LOADM      IC     3    0380   0089     8    7  KWN
 LOADM.C    T      2    0100   008B    11    7  KWN
 LOADM.D    T     16    04C0   008A     4   14  KWN
 MAZE       05     4    04C0   007A     0    8  KWN
 NOTICES    T      8    04C0   0042     0   73  KWANEU
 NUMSTR     05    40    04C0   0098     4   21  KWN
 PATCH1.5   IS     4    04C0   0079    12    7  KWN
 PERMTEST   05     4    04C0   007B     4    8  KWN
 PERSONAL   06    12    04C0   0051     8    4  KWN
 PHONUM:D   T      4    04C0   008F     0   17  KWN
 PHONUM:S   05     8    04C0   008E     4   15  KWN
 PRIMES     05     4    04C0   007D     0   11  KWN
 QUIKSORT   05     4    04C0   0083     2   13  KWN
 ROBOTS     05    16    04C0   0092     4   17  KWN
 ROBOTS:C   T      1    0100   0094    15   13  KWN
 SER.RNDM   05     4    04C0   0052     4    5  KWN
 SERIAL     05     4    04C0   0053     8    5  KWN
 SHELSORT   05     4    04C0   0084     6   13  KWN
 SORT       05     4    04C0   0073     8    1  KWN
 STR-SORT   05     8    04C0   009C     8   29  KWN
 TENSOR:C   T      1    0100   0095     4   20  KWN
 TENSORTS   05    28    04C0   0093     8   18  KWN
 WARRANTY   T      4    04C0   0040     4   72  KWANEU
 WK.DOC     T     28    04C0   0099    12   23  KWN
 build      $      2    0100   0081    14   12  KWN
 mes.s      T      8    04C0   007E     4   11  KWN
 mes.temp   T      2    0100   0082     0   13  KWN
 message    I      1    0100   007F    12   11  KWN
 text       T      4    04C0   0088     8   15  KWN
```

CONTENTS OF HELIOS LIBRARY H8

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

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

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

```
04/16/81  FILES ON: H-8

  NAME      TYPE  SIZE    BLKZ     ID     SEC   TRK  ATTRI    INDEX
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 C80LIB.A    T      32    09C0    0017     8     7   KWN
 CC          Ic     80    09C0    0046     0    48   KWN
 CC.DEF      T       8    04C0    0027     9     1   KWN
 CONTENTS    T       4    04C0    004A     4    57   KWANEU
 Cc.A        T       1    0100    0026    15     0   KWN
 Cc1.A       T      56    09C0    0015     0     1   KWN
 Cc1.C       T      56    09C0    0016     0     4   KWN
 Cc2.A       C      48    04C0    0043     0     7   KWN
 Cc2.C       T      48    09C0    001C     0    20   KWN
 Cc3.A       T      48    09C0    001B     8    17   KWN
 Cc3.C       T      32    09C0    001D     8    22   KWN
 Cc4.A       T      56    09C0    0019     8    11   KWN
 Cc4.C       T      32    09C0    0021     8    32   KWN
 Cc5.A       T      48    09C0    001E     0    24   KWN
 Cc5.C       T      24    04C0    0030    12    42   KWN
 Cc6.A       T      40    09C0    001F     8    27   KWN
 Cc6.C       T      24    04C0    0034    12    44   KWN
 Cc7.A       T      48    09C0    0020     0    30   KWN
 Cc7.C       T      24    04C0    0035     4    46   KWN
 Cc8.A       T      32    09C0    001A     8    14   KWN
 Cc8.C       T      32    09C0    0025     8    38   KWN
 FEEDBACK    T       8    04C0    004B     4     7   KWANEU
 WARRANTY    T       4    04C0    004C    12    57   KWANEU
 XASSM       I.     28    04C0    0048    12    47   KWN
 cc.txt      T      52    04C0    003A     0    37   KWN
 cc.use      T       4    04C0    0047     8    40   KWN
 setup.a     C      12    04C0    0044    12    35   KWN
 test.c      T       1    0100    003E     8     1   KWN
```

— BOOK REVIEW —
by Jeff Tom

SOFTWARE DEBUGGING FOR MICROCOMPUTERS
Robert C. Bruce $17.95
ISBN 0-8359-7021-3 or 0-8359-7020-5 pbk

I was wandering around near the University of Arizona in Tucson when I stopped to look in the window of this bookstore and there it was. I went in to see if it pertained to 8080 microprocessors, and found that it was for debugging programs written in BASIC. But wait...what is this?? Processor Technology Extended Cassette Basic? Yes! For that reason alone, I bought the book. Was it worth it? Read on.

First let me say that the book is not intended to be a tutorial on BASIC, but any newly presented, or unique statements are briefly described before being used. On the other hand, one could probably learn the function of many of the statements by seeing them used in the programs if nothing else.

I found the approach of the book to be very much in line with human nature. The author "writes" a program, and then when it doesn't work, he starts to debug it, often resorting to the user's manual when all else fails. How many of us haven't done that at one time or another?

The book doesn't present any earth-shaking plan for writing error-free code, but stresses the use of flowcharts, modular programming, PRINT statements, and playing computer with paper and pencil. He does recommend using simple numbers like 0 and 1 where possible to simplify catching math errors, which makes good sense to me. He encourages the use of REM statements and indented FOR-NEXT loops, but there are places in the programs where REM statements are few and far between, and the only place his loops are indented is when he shows how nice it makes tracing program flow. This was unusual because somehow, E.C. BASIC's "print pretty" feature had been defeated.

About half the book is devoted to developing a data base management system, and these chapters deal rather well with the subjects of string manipulation, and use of cassette files.

Few of the programs presented are anything that you'd run to your computer to type in, but most could be adapted for personal use with a little thought. As a matter of fact, changing programs from one use to another is one of the subjects covered, along with some of the pitfalls that can be encountered.

I won't guarantee that you'll learn anything new from this book, but it does make you think about your programming practices and it may point out a trick or two for you to use. One thing is fairly certain, this is the first and last book to be devoted to E.C. BASIC.

COMPUTER COURSES FOR THE DEAF

Rochester Institute of Technology (RIT) will offer two computer courses for deaf adults this summer through the National Technical Institute for the Deaf (NTID).

Introduction to Data Processing - August 3 - 7
Advanced Data Processing   -   August 10 - 14

For more information, contact Donald Beil, NTID Data Processing Dept., Rochester Institute of Technology, One Lomb Memorial Dr., Rochester, NY 14623 or (716)475-6373.

Repairing your Sol (Part 2)

by Joe Maguire

In the previous installment of this series, we concentrated on isolating the problem area. We left off just as we were going to tackle the Sol PC board.

The operation of the Sol can be divided into a number of functional blocks. From the point of view of the operator they are:

1. The video display
2. The CPU
3. The cassette tape I/O
4. The serial port
5. The parallel port
6. The SOLOS ROM
7. The C8ØØ-CBFF memory
8. The keyboard

Beg, borrow or steal a copy of the Sol PC block diagram from the Sol manual. (X-24) This diagram has each of the ICs associated with the above blocks listed in tabular form. It's possible to proceed without it but it's going to be tougher.

Now is the time to ask the PC, "where does it hurt?" In other words, what isn't working properly. Probably the most serious fault is no video. Without that to assist in trouble shooting you are groping in the dark. The first thing to check here is the monitor. The best way to check it is to try it on another computer. This is where friends and computer clubs really become valuable! When taking your monitor to your friend's, be sure to take your video connector cable along too. I've never found a bad monitor but I've found many shorted cables. Assuming the monitor is OK, we now start the real search for the problem area.

A completely dead Sol (one with no CPU activity) will not generate any video. So how can we tell whether to start with the video circuits or the CPU logic? Try some commands from SOLOS which will produce a response outside the computer. For example: Can you boot up your disks? Can you SAVE something on a cassette tape? If it seems like these activities are working but the screen remains blank, it's most likely the VDM. If nothing happens, you've got a dead one.

Assuming that the proper voltages are arriving from the power supply, the most likely cause of death would be no clock signal. The Sol is a digital computer, which means, all circuits within its innards must march in step with the "drummer". If the drummer (crystal oscillator) gets tired, everybody else takes a break too. About the only way to check if the crystal and clock circuits are working is with a logic probe or an oscilloscope. A rough idea can be had by holding a small transistor radio near the PC board. If there is any clock activity, the radio will produce all sorts of whines and buzzes.

Third Step, Repair

I'm going to let you in on a secret. I'm going to tell you how the "experts" find the bad ICs. Now I know you are visualizing racks of test equipment: oscilloscopes, digital analyzers, logic emulators and the like. But that's not the secret. Oh, those things are nice for quickly getting to the bad functional area but, when it gets down to picking out the bad IC, do you know how the "experts" do it? I'll tell you. They get another Sol that's working and start swapping ICs until the trouble disappears! The great thing about this method is that you don't need any test equipment. The absolutely worst case would be that you had to swap every IC on the PC board. If you can narrow down the problem area even a little you can have your Sol up and running in no time. Now you know the measure of a real friend. He is one who will lend you his Sol for testing!

As I said, the worst problem is no video. There are 29 ICs associated with the VDM circuit and here is where that page X-24 from the Sol manual can really save you some time. It lists all of them in a neat group. Start swapping until the video returns to normal. I recommend swapping one at a time, of course turning the power off between each change. Two other conditions can cause strange video displays so if all 29 ICs have been swapped and the problem isn't cured, it could be one of them.

The VDM in the Sol is known as a "memory mapped" display circuit. This means that the characters shown on the screen are actually stored in a section of the computer's memory. In the Sol, this video memory resides between addresses CCØØ and CFFF hex. Eight memory ICs make up this video block. (U14-U21) If one of them goes bad, the characters cannot be formed properly and you get a crazy looking display. A memory test of this 1K block should turn up the bad chip. Have you ever tested the video memory? It gives a wild display! The other thing which can cause funny characters to appear on the VDM is a bad keyboard. There are two ICs on the keyboard which are very sensitive to static charges. (U19,U22) I have found a number of these bad and when they go, all the wrong characters are sent out. Try exchanging keyboards from the test Sol to see if this is the problem.

The CPU can be checked by giving any of the SOLOS commands. If even one works OK then the CPU is probably not at fault. Look suspiciously at the SOLOS ROM instead. Of course, if you are trying a tape command or the TERM command then the problem might be with the tape I/O circuit or the serial UART. A program that runs amuck frequently may indicate a bad memory IC in the SOLOS RAM area. This 1K memory block (C8ØØ-CBFF) is often used for the program stack and only one bad bit can send the CPU off into never never land. If you can't get any SOLOS command to work or even get the prompt to appear, this is a good place to suspect trouble as all SOLOS commands use this memory for stack purposes. CPU support ICs can cause problems such as failure to jump to SOLOS on RESET, bad address or data signals and some really weird symptoms. This is another area to check if nothing will work.

The cassette tape circuit as well as the serial and parallel ports can best be checked by connecting a device, known to be working properly, to their respective connectors and running some test programs. Things to look for if you suspect too much voltage got on the printer cable would be the line driver ICs for the serial port, (U38,U56) or the six ICs associated with parallel I/O. The cassette tape motor relays are known to get stuck occasionally and when this happens they usually need to be replaced.

Barring a catastrophic accident, (like being hit by lightning) changing ICs should find most of the problems. Shorted circuit traces, bad IC sockets and the like of those ills generally will not be found in a computer which has been operating sucessfully for some time. Those are the bane of kit builders.

In the next issue: The memory boards.

...DEAD KEYS ON MY SOL

I would appreciate it if someone would explain what to do about dead keys on the Sol keyboard.

Emile Roth, 1001 Evelyn Terrace East, #104, Sunnyvale, CA 94086

(Editor's note: The dead keys are usually due to a bad foam pad inside the key plunger. The keytops pull straight off. Beneath, will see the mechanism which is screwed down to the keyboard. Remove the mechanism and inside you'll find a cylindrical piece of foam plastic with a self-adhering layer of metal foil on it. The foil is pressed down against the printed-circuit pads to make the k register in the circuit. Old pads fail to make good proximity. Keytronics, the manufacturer of the keyboard, will only sell the pads in large quantity (unless you happen to get to a salesperson with a soft heart that day).
IF ANYONE HAS EXTRA FOAM KEYPADS, PLEASE CONTACT EMILE.
--Stan.)

```
*
*  North Star PASCAL Input/Output Routine
*
*  This I/O routine is for use with North
*  Star Pascal Ver. 1.0 and a Sol computer.
*
*  Written by: Stephen Maguire    July, 1980
*              P.O. Box 3742 DT
*              Anchorage, AK 99510
*
*  It provides the following support:
*
*  a)  It correctly interprets and performs
*      the GOTOXY procedure that comes with
*      the system so that BINDER need never
*      be used.  This allows instant cursor
*      positioning without the need to write
*      the "necessary" GOTOXY procedure de-
*      scribed in the manual.
*
*  b)  It supports PRINTER: so that output
*      can be printed out.  The routine is
*      for an NEC Spinwriter 5510, but will
*      work for any serial printer if wired
*      according to protocol shown below.
*
*      Spinwrtr pin    Sol Serial pin
*
*      TX DATA   2      3   RX DATA
*      RX DATA   3      2   TX DATA
*          GND   7      7   GND
*          CTS   5|
*          DSR   6|     20  DTR  | denotes common
*          CD    8|               connection
*                       |6  DSR
*          DTR  20      |8  CD
*      REV CHA  19      5   CTS   See Note below:
*
*      Note: In this driver, the Reverse Channel
*      pin of the Spinwriter is used in the "LOW"
*      mode by setting #5 of SW1 to "ON" (up) on
*      the control panel circuit board. (G9BNF)
*      This results in a "high" to the Sol when
*      characters can be accepted. If the printer
*      is unplugged or turned off, the Sol still
*      sees a high because of its own internal
*      circuitry and will continue sending char-
*      acters. This prevents a program hang (and
*      possibly a crash) if the printer is not
*      available.
*
*  c)  If input is asked of the printer, input
*      from the keyboard is checked for instead.
*
*  d)  If a control-p is sent to CONSOLE:, output
*      to CONSOLE: is sent to PRINTER: instead.
*      This continues until another control-p is
*      encountered.  (The control-p may be typed
*      at the keyboard or output in a program.)
*
*  e)  On initialization, the memory is sized and
*      then waits for either a carriage return
*      or a hex value to set the memory limit.
*      This allows the user to "protect" high
*      memory if necessary.
*
*  f)  The underline character can be printed
*      in order to facilitate compatibility
*      with other systems.
*
```

```
        *  g)  Control-L erases to end of line.
        *
        *  h)  The bell character is sent to PRINTER:
        *      instead of to CONSOLE:.
        *
        *  org 2400H - SYSTEM.NSTAR2  (memory at 2000H)
        *  org 400H - SYSTEM.NSTAR0  (memory at 0000H)
        *
0400             ORG    400H
        *
00FF    TRUE     EQU    0FFH
0000    FALSE    EQU    0
        *
0007    CHBEL    EQU    07H      The bell
000A    CHLFE    EQU    0AH      Linefeed
000B    CHCLR    EQU    0BH      CLEAR screen character
000C    CHFFD    EQU    0CH      Formfeed
000D    CHCR     EQU    0DH      Carriage return
000E    CHOME    EQU    0EH      HOME CURSOR character
0010    CHDLE    EQU    10H      Control-p
001B    CHESC    EQU    1BH      ESCAPE character
005F    CHUND    EQU    5FH      underline character
007F    CHDEL    EQU    7FH      The DEL character
007F    ASCII    EQU    7FH      largest ASCII
        *
        *  Equates determined by STANDARD SOLOS
        *
C019    SOLOUT EQU    0C019H  Solos output routine
C098    OCHAR  EQU    0C098H  print an underline
C0F4    CLINE  EQU    0C0F4H  erase to end of the line
C11C    VDADD  EQU    0C11CH  calculate screen address
C340    SHEX   EQU    0C340H  convert ASCII to binary
C40B    HEOUT  EQU    0C40BH  print register A in ASCII
C808    NCHAR  EQU    0C808H  X coordinate of cursor
C809    LINE   EQU    0C809H  Y coordinate of cursor
C80A    BOT    EQU    0C80AH  Text offset
        *
        *  CONSOLE: device (keyboard)
        *
00FA    CSTAT    EQU    0FAH     Keyboard status port
00FC    CDATA    EQU    0FCH     Keyboard data port
0001    CRDYINP EQU    1
        * CRDYOUT EQU 2  (Not used)
        *
        *  PRINTER: (no input -- CONSOLE: input
        *            is used instead)
        *
00F8    PSTAT    EQU    0F8H     Printer status port
00F9    PDATA    EQU    0F9H
        * PRDYINP EQU 2  (Not used)
00A0    PRDYOUT EQU    0A0H    "high" on CTS and TBE pins
        *
        *  REMOTE:  device (not supported)
        *
        * RSTAT EQU 6   (For optional extra device.
        * RDATA EQU 0   Values are from the sample I/O
        * RRDYINP EQU 2   routine given in the manual.
        * RRDYOUT EQU 1   OK for tape I/O if the
        * RSTROBE EQU 80H   required code is written.)
        * RPOFLG EQU 20H
        *
        *
0009    NOTRDY EQU     9        Not-ready value
00E8    DCTRLB EQU     0E8H     High byte of standard PROM
0080    DDENS  EQU     80H      Double density
0000    SDENS  EQU     0        Single density
0000    ONESIDE EQU    0        For single sided drives
0040    TWOSIDE EQU    40H      Quad capacity drives
0005    SBLKTRK EQU    5        blocks/track in single-dens
000A    DBLKTRK EQU    10       blocks/track in double-dens
```

```
                *
                * * * * * * * * * * * * * * * * * * * * * * *
                * Set line below according to your drives *
    00CA        CHARACS EQU   TWOSIDE+DDENS+DBLKTRK  (Quad) *
                * * * * * * * * * * * * * * * * * * * * * * *
    0400        NSJTST EQU    $
    0600        STRTSR EQU    NSJTST+512
                *
                *  CONSOLE:  routine addresses
                *
0400 C3 2E 05   CONOCL JMP    ONLINE  Keyboard is always "ready"
0403 C3 2D 04          JMP    CONINP  Keyboard in
0406 C3 3D 04   CONESC JMP    CONOUT  Video out
0409 C3 F1 04          JMP    CONST
                *
                *  PRINTER:  routine addresses
                *
040C C3 2E 05   PTRONL JMP    ONLINE  Keyboard is always "ready"
040F C3 2D 04          JMP    CONINP  If input, go to CONSOLE: in
0412 C3 20 05          JMP    PRNT2
                *
                *  REMOTE:  routine addresses
                *
0415 C3 30 05   REMONL JMP    OFFLIN  Offline (not supported)
0418 C3 FB 04          JMP    REMINP
041B C3 FB 04          JMP    REMOUT
                *
041E C3 33 05          JMP    NSMSIZ  How much memory is available?
0421 C3 30 05          JMP    OFFLIN  No system clock, it is offline
0424 C3 38 05          JMP    MACINT  The initialization routine
                *
0427 CA         DV4CHR DB     CHARACS  Device characteristics
0428 CA         DV5CHR DB     CHARACS  (set for your drives)
0429 CA         DV9CHR DB     CHARACS
042A CA         DV10CHR DB    CHARACS
                *
042B 00 00      EXPANSN DW    0       For future use
                *
                *  CONSOLE: input  (PRINTER: input)
                *
042D DB FA      CONINP IN     CSTAT   Has a key been typed?
042F 2F                CMA    .       Inverse the value
0430 E6 01             ANI    CRDYINP  Strip the value
0432 CA 2D 04          JZ     CONINP  No, keep waiting
0435 DB FC             IN     CDATA   Yes, get the character
0437 FE 10             CPI    CHDLE   Has a control-p been entered
0439 C0                RNZ    .
043A F6 80             ORI    80H     Ctrl-p's have high bit set
043C C9                RET    .
                *
                *  CONSOLE: output
                *
043D 79         CONOUT MOV    A,C     Get the character
043E E6 7F             ANI    ASCII
0440 4F                MOV    C,A     Save the stripped value
                *
0441 FE 07             CPI    CHBEL   If bell, send to printer
0443 CA 20 05          JZ     PRNT2
                *
0446 FE 1B             CPI    CHESC   Escape says GOTOXY
0448 C2 57 04          JNZ    CON0
044B CD 1C C1          CALL   VDADD
044E 7E                MOV    A,M     Remove the cursor
044F C6 80             ADI    80H
0451 77                MOV    M,A
0452 3E 01             MVI    A,1     Set flag to indicate so
0454 C3 AB 04          JMP    GOTO3
                *
0457 FE 10      CON0   CPI    CHDLE   ctrl-p, toggle
0459 C2 66 04          JNZ    CON1
```

```
045C 3A FC 05          LDA    TOGGLE  Get the switch
045F C6 80             ADI    80H     Now flip it
0461 32 FC 05          STA    TOGGLE
0464 AF                XRA    A
0465 C9                RET
                *
0466 3A FB 05   CON1   LDA    XYDATA  Is it GOTOXY?
0469 B7                ORA    A
046A CA B0 04          JZ     CON2    Yes, it is
                *
                *  GOTOXY makes "ESC","=",y,x to screen address
                *
046D FE 01      GOTOXY CPI    1       ESCAPE has been received,
046F C2 7F 04          JNZ    GOTO0   Now check for the "=" sign
0472 79                MOV    A,C
0473 FE 3D             CPI    '='
0475 CA A5 04          JZ     GOTO2   Yes, its the "=" sign
0478 AF                XRA    A       Error, abort GOTOXY procedure
0479 32 FB 05          STA    XYDATA
047C C3 3D 04          JMP    CONOUT  And output the Character
                *
047F FE 02      GOTO0  CPI    2       Calculate the row value
0481 C2 8F 04          JNZ    GOTO1
0484 79                MOV    A,C     Get the row value
0485 DE 20             SBI    32      Sub 20H to get correct value
0487 E6 0F             ANI    0FH     Make sure value in range
0489 32 09 C8          STA    LINE
048C C3 A5 04          JMP    GOTO2
                *
048F 79         GOTO1  MOV    A,C     Calculate column value
0490 DE 20             SBI    32      Subtract the offset
0492 E6 3F             ANI    3FH     Make sure value in range
0494 32 08 C8          STA    NCHAR   Store the value
0497 CD 1C C1          CALL   VDADD   Calculate screen address
049A 7E                MOV    A,M     Save character at cursor pos
049B F6 80             ORI    80H     Put the cursor there
049D 77                MOV    M,A     This does it
                *
049E 3A FC 05          LDA    TOGGLE  If print on, output CR/LF
04A1 B7                ORA    A
04A2 C4 11 05          CNZ    PRNT0
                *
04A5 3A FB 05   GOTO2  LDA    XYDATA  Increment pointer so we
04A8 3C                INR    A       know which argument to get.
04A9 E6 03             ANI    3
                *
04AB 32 FB 05   GOTO3  STA    XYDATA  Save the value
04AE AF                XRA    A
04AF C9                RET    .
                *
                *  All done with the GOTOXY procedure
                *
04B0 3A FC 05   CON2   LDA    TOGGLE  If PRINTER: is on,
04B3 B7                ORA    A       send it the output
04B4 C2 FC 04          JNZ    PRINT
                *
04B7 79                MOV    A,C
04B8 FE 0C             CPI    CHFFD   Formfeed character?
04BA C2 CB 04          JNZ    CON3    No, so go print character
04BD E5                PUSH   H       HL cannot be destroyed
04BE CD 1C C1          CALL   VDADD
04C1 E5                PUSH   H       Save this screen address
04C2 CD F4 C0          CALL   CLINE   Call erase
04C5 E1                POP    H       Get it back
04C6 36 A0             MVI    M,0A0H  Put on the cursor
04C8 E1                POP    H
04C9 AF                XRA    A
04CA C9                RET    .
                *
04CB FE 7F      CON3   CPI    CHDEL   Delete?, put cursor-left
```

```
04CD C2 D2 04        JNZ    CON4
04D0 3E 01           MVI    A,1
                *
04D2 FE 5F      CON4 CPI    CHUND  allow underline
04D4 C2 E7 04        JNZ    CON5
04D7 E5              PUSH   H      Save these registers
04D8 C5              PUSH   B
04D9 47              MOV    B,A    Put the character in B
04DA CD 98 C0        CALL   OCHAR  Output the character
04DD CD 1C C1        CALL   VDADD  Now, put on the cursor
04E0 7E              MOV    A,M    Do it now
04E1 F6 80           ORI    80H
04E3 77              MOV    M,A
04E4 C3 ED 04        JMP    CON6   Exit gracefully
                *
04E7 E5         CON5 PUSH   H      Save all registers,
04E8 C5              PUSH   B
04E9 47              MOV    B,A    Put the character in B
04EA CD 19 C0        CALL   SOLOUT  Now, put to screen
                *
04ED C1         CON6 POP    B      Restore the registers
04EE E1              POP    H
04EF AF              XRA    A
04F0 C9              RET
                *
                *  CONSOLE: input status
                *
04F1 DB FA      CONST IN    CSTAT
04F3 2F              CMA    .
04F4 E6 01           ANI    CRDYINP
04F6 3E 00           MVI    A,FALSE
04F8 C8              RZ     .      Return now if "not ready"
04F9 3E FF           MVI    A,TRUE
                *
                *  REMOTE:  not implemented so do RETURN
                *
    04FB        REMIN  EQU  $
04FB C9         REMOUT RET  .
                *
                *  PRINTER:  output (via control-p toggle)
                *
04FC 79         PRINT MOV   A,C    Get the character
04FD FE 0D           CPI    CHCR
04FF CA 20 05        JZ     PRNT2
0502 FE 0A           CPI    CHLFE
0504 CA 20 05        JZ     PRNT2
0507 FE 0E           CPI    CHOME  Home?, print C/R, LF
0509 CA 11 05        JZ     PRNT0
050C FE 0B           CPI    CHCLR  Clear also
050E C2 1B 05        JNZ    PRNT1
                *
0511 0E 0D      PRNT0 MVI   C,CHCR  Print it
0513 CD FC 04        CALL   PRINT
0516 0E 0A           MVI    C,CHLFE
0518 C3 FC 04        JMP    PRINT
                *
051B FE 20      PRNT1 CPI   20H
051D DA 2E 05        JC     ONLINE  Control char?, ignore it
                *
                *  PRINTER:  output (standard output routine)
                *
0520 DB F8      PRNT2 IN    PSTAT
0522 E6 A0           ANI    PRDYOUT
0524 FE A0           CPI    PRDYOUT
0526 C2 20 05        JNZ    PRNT2  Loop until ready
0529 79              MOV    A,C
052A E6 7F           ANI    ASCII
052C D3 F9           OUT    PDATA
                *
052E AF         ONLINE XRA  A
```

```
052F C9              RET
                *
0530 3E 09      OFFLIN MVI  A,NOTRDY
0532 C9              RET
                *
0533 2A FA 05   NSMSIZ LHLD MEMORY  Get memory size
0536 E3              XTHL
0537 E9              PCHL   Return now
                *
                *  Boot-up initialization
                *
0538 21 B3 05   MACINT LXI  H,MES1  Print start message
053B CD A0 05        CALL   PRASC
                *
053E 21 FD 05        LXI.   H,ENDMARK  Start sizing at end
0541 3E AA           MVI    A,0AAH  Test byte
0543 46         SIZE MOV    B,M    Save this memory location
0544 36 AA           MVI    M,0AAH  Put in test byte
0546 BE              CMP    M      OK? If not, ROM or no memory
0547 70              MOV    M,B    Put the old value back
0548 23              INX    H      Move to the next location
0549 CA 43 05        JZ     SIZE
                *
054C 2B              DCX    H      Point to last good location
054D 2B              DCX    H
054E 7D              MOV    A,L    Make HL an even number
054F E6 FE           ANI    0FEH
0551 6F              MOV    L,A
0552 EB              XCHG   .      DE has last memory location
                *
0553 21 DC 05        LXI    H,MES2  Print this message
0556 CD A0 05        CALL   PRASC
0559 CD AB 05        CALL   PRHEX  Now, print the value
055C 0E 20           MVI    C,' '  Print a following blank
055E CD 3D 04        CALL   CONOUT
                *
0561 CD 1C C1        CALL   VDADD  Get the cursor address
0564 E5              PUSH   H      Save this
0565 06 00           MVI    B,0    So far, no characters
0567 CD 2D 04   DIFSIZE CALL CONIN  Check for input
056A CA 67 05        JZ     DIFSIZE  None yet
056D FE 0D           CPI    CHCR   Carriage return?
056F CA 87 05        JZ     DIF4   Terminate input
0572 FE 7F           CPI    CHDEL  Delete?
0574 C2 80 05        JNZ    DIF3   No, much be a hex number
                *
0577 78              MOV    A,B    If no characters typed,
0578 B7              ORA    A
0579 CA 67 05        JZ     DIFSIZE  Then don't back up
057C 05              DCR    B      Else adjust character count
057D 05              DCR    B      (for later adjustment)
057E 3E 5F           MVI    A,CHDEL-20H  Output back-up
                *
0580 04         DIF3 INR    B
0581 CD E7 04        CALL   CON5   allow delete
0584 C3 67 05        JMP    DIFSIZE  Go get more input
                *
0587 CD 1C C1   DIF4 CALL   VDADD  Get cursor location
058A 36 20           MVI    M,' '  Erase the cursor
058C E1              POP    H      Get start location of hex
058D 78              MOV    A,B    Any numbers been entered?
058E B7              ORA    A
058F CA 9B 05        JZ     DIF5   If not, use calculated value
0592 EB              XCHG   .      Otherwise,
0593 CD 40 C3        CALL   SHEX   Convert
0596 7D              MOV    A,L    Make sure the value is even
0597 E6 FE           ANI    0FEH
0599 6F              MOV    L,A
059A EB              XCHG   .      Put the value in DE
                *
```

```
Ø59B EB             Ø42Ø DIF5   XCHG   .      This puts DE in HL
Ø59C 22 FA Ø5       Ø421        SHLD   MEMORY  Save it for later
Ø59F C9             Ø422        RET
                    Ø423 *
                    Ø424 *  Message print routine
                    Ø425 *
Ø5AØ AF             Ø426 PRASC  XRA    A      This tests end-of-msg
Ø5A1 BE             Ø427        CMP    M      At end?
Ø5A2 C8             Ø428        RZ     .      If so, return immediately
Ø5A3 4E             Ø429        MOV    C,M    Get the character
Ø5A4 CD 3D Ø4       Ø43Ø        CALL   CONOUT  Send it out
Ø5A7 23             Ø431        INX    H      Bump pointer to next
Ø5A8 C3 A1 Ø5       Ø432        JMP    PRASC+1  keep looping
                    Ø433 *
                    Ø434 *  PRHEX print value of DE as ASCII
                    Ø435 *
Ø5AB 7A             Ø436 PRHEX  MOV    A,D    Print D
Ø5AC CD ØB C4       Ø437        CALL   HEOUT
Ø5AF 7B             Ø438        MOV    A,E    Print E
Ø5BØ C3 ØB C4       Ø439        JMP    HEOUT
                    Ø44Ø *
                    Ø441 *  Messages and data storage
                    Ø442 *
Ø5B3 ØB             Ø443 MES1   DB     CHCLR
Ø5B4 55 43 53 44    Ø444        ASC    'UCSD Pascal for Solos'
     2Ø 5Ø 61 73
     63 61 6C 2Ø
     66 6F 72 2Ø
     53 6F 6C 6F
     73
Ø5C9 ØD ØA          Ø445        DB     CHCR,CHLFE,CHLFE
Ø5CC 53 69 7A 69    Ø446        ASC    'Sizing memory'
     6E 67 2Ø 6D
     65 6D 6F 72
     79
Ø5D9 ØD ØA          Ø447        DB     CHCR,CHLFE,Ø
                    Ø448 *
Ø5DC 4D 65 6D 6F    Ø449 MES2   ASCZ   'Memory available to address: '
     72 79 2Ø 61
     76 61 69 6C
     61 62 6C 65
     2Ø 74 6F 2Ø
     61 64 64 72
     65 73 73 3A
     2Ø ØØ
Ø5FA ØØ ØØ          Ø45Ø MEMORY DW     Ø
Ø5FC ØØ             Ø451 XYDATA DB     Ø
Ø5FD ØØ             Ø452 TOGGLE DB     Ø
                    Ø453 *
     Ø5FE           Ø454 ENDMARK EQU   $
                    Ø455 *
                    Ø456        END
```

EXTENSIONS FOR SOFTWARE # 1: FILE RENUMBERING

Software #1 is a complete Editor-Assembler-Monitor package which was distributed by Processor Technology. It was produced in at least two versions that I know of. The earliest version was a true self-contained system that included drivers for a terminal interface. This version was distributed with both source and object code back in the days when PTCo did not yet make a complete computer system. A later version was distributed as a CUTS tape (object only). This one took advantage of many features of the CUTER/SOLOS monitor, including the tape interface. I suspect that ALS-8 may be basically an enhancement of Software #1--in any case they both use the same kind of file structure, in which lines are numbered and each line is preceded by a byte count.

I obtained the original version of Software #1 some time ago, before I decided to make my homebrew system into a pseudo-Sol. Because it fit into 4K of RAM, could be adapted for ROM, and was well documented I made much use of the package, modified it extensively, and learned how it worked internally. Many of my modifications would be hard to apply to the more well-known cassette version, but I have rewritten a few of them to apply to this version. I would like to make a few of these available to PROTEUS members.

The following program renumbers a Software #1 source file. The first line is numbered 0010 and succeeding lines are spaced 10 numbers apart. This is a big step toward making Software #1 easy to use for serious program development.

The listing that follows has an odd origin because it resides in my system immediately below a 6502 cross-assembler that sits in the 1800H to 1FFFH region and. This cross-assembler is a parasite on Software #1 which uses every possible subroutine in the PTCo package. I didn't want to tamper with Software #1 itself, so I let it continue to use the area from 0F60H up for the assembler symbol table. This is why my extensions are crowded into the top of the 1st 8K of memory. Since Software #1 is a memory-to-memory assembler it is unlikely that any programs to be assembled will be so large they will require more than 1K for the symbol table.

Albert S. Woodhull
RFD 2
Amherst, Mass., 01002

```
FILE
NUMB  2000  2AAF

ASSM 1754 3000


1754                   0010 ;NUMB
1754                   0020 ;TO RENUMBER SOURCE FILE
1754                   0030 ;       A.S.WOODHULL
1754                   0040 ;       7/VII/79
1754                   0050 ;      edited 25 Jan 81
1754                   0060 ;
1754                   0070 ;TEST FOR CURRENT FILE
1754 CD 00 18          0080 NUMB    CALL CFTST
1757                   0090 ;
1757                   0100 ;1ST CHECK FOR LESS THAN 999 LINES
1757 11 00 00          0110 NUM     LXI D,0
175A 2A 58 0D          0120         LHLD BOFP
175D                   0130 ;
175D                   0140 ;GET COUNT,TEST FOR EOF
175D 7E                0150 CNT     MOV A,M
175E FE 01             0160         CPI 1
1760 CA 79 17          0170         JZ DONUM
1763                   0180 ;
1763                   0190 ;POINT TO NEXT COUNT
1763 85                0200         ADD L
1764 6F                0210         MOV L,A
1765 7C                0220         MOV A,H
1766 CE 00             0230         ACI 0
1768 67                0240         MOV H,A
1769                   0250 ;COUNT THAT LINE
1769 13                0260         INX D
176A                   0270 ;LESS THAN 999 LINES?
176A 7A                0280         MOV A,D
176B FE 03             0290         CPI 3
176D DA 5D 17          0300         JC CNT
1770 7B                0310         MOV A,E
1771 FE E8             0320         CPI 0E8H
1773 DA 5D 17          0330         JC CNT
1776                   0340 ;QUIT IF TOO MANY LINES
1776 C3 F1 17          0350         JMP ABORT
1779                   0360 ;
1779                   0370 ;INITIALIZE LINE NO,GET SOF
```

```
1779 21 30 30         0380 DONUM  LXI H,3030H
177C 22 BF 0D         0390        SHLD LNLO
177F 22 BD 0D         0400        SHLD LNHI
1782 2A 58 0D         0410        LHLD BOFP
1785 EB               0420        XCHG
1786                  0430 ;
1786 1A               0440 NLOOP  LDAX D          ;GET LINE LENGTH
1787 FE 01            0450        CPI 1           ;CK FOR END OF FILE
1789 CA E2 17         0460        JZ EOF
178C                  0470 ;POINT TO LSD OF LINE NO.
178C 13               0480        INX D
178D 13               0490        INX D
178E 13               0500        INX D
178F 13               0510        INX D
1790                  0520 ;REPLACE OLD LINE NUMBERS
1790 3E 30            0530        MVI A,30H       ;ASCII ZERO
1792 12               0540        STAX D
1793 3A BF 0D         0550        LDA LNLO        ;GET 10'S DIGIT
1796 3C               0560        INR A
1797 FE 3A            0570        CPI 3AH         ;DECIMAL CY TEST
1799 D2 A7 17         0580        JNC CY10
179C 1B               0590        DCX D
179D 12               0600        STAX D
179E 32 BF 0D         0610        STA LNLO
17A1 3A BE 0D         0620        LDA LNHI+1      ;GET 100'S DIGIT
17A4 C3 B7 17         0630        JMP DOC
17A7                  0640 ;IF CARRY OUT OF 10'S DO NEXT
17A7 3E 30            0650 CY10   MVI A,30H
17A9 1B               0660        DCX D
17AA 12               0670        STAX D
17AB 32 BF 0D         0680        STA LNLO
17AE 3A BE 0D         0690        LDA LNHI+1      ;GET 100'S
17B1 3C               0700        INR A
17B2 FE 3A            0710        CPI 3AH         ;TST FOR DEC CARRY
17B4 D2 C2 17         0720        JNC CYC
17B7 1B               0730 DOC    DCX D           ;STORE 100'S
17B8 12               0740        STAX D
17B9 32 BE 0D         0750        STA LNHI+1
17BC 3A BD 0D         0760        LDA LNHI        ;GET 1000'S
17BF C3 D2 17         0770        JMP DOK
17C2                  0780 ;DO NEXT IF CARRY OUT OF 100'S
17C2 3E 30            0790 CYC    MVI A,30H
17C4 1B               0800        DCX D
17C5 12               0810        STAX D
17C6 32 BE 0D         0820        STA LNHI+1
17C9 3A BD 0D         0830        LDA LNHI        ;GET 1000'S
17CC 3C               0640        INR A
17CD FE 3A            0850        CPI 3AH
17CF D2 F1 17         0860        JNC ABORT
17D2 1B               0870 DOK    DCX D           ;STORE 1000'S
17D3 12               0880        STAX D
17D4 32 BD 0D         0890        STA LNHI
17D7                  0900 ;
17D7                  0910 ;NOW COMPUTE START OF NEXT LINE
17D7 1B               0920 NEXT   DCX D
17D8 1A               0930        LDAX D
17D9 83               0940        ADD E
17DA 5F               0950        MOV E,A
17DB 7A               0960        MOV A,D
17DC CE 00            0970        ACI 0
17DE 57               0980        MOV D,A
17DF C3 86 17         0990        JMP NLOOP
17E2                  1000 ;
17E2                  1010 ;WHEN DONE PUT NEW MAXLINE IN DIR
17E2 2A BD 0D         1020 EOF    LHLD LNHI
17E5 22 5C 0D         1030        SHLD MAXL
17E8 2A BF 0D         1040        LHLD LNLO
17EB 22 5E 0D         1050        SHLD MAXL+2
```

```
17EE C3 16 00         1060        JMP RDY
17F1                  1070 ;
17F1                  1080 ;ONLY ALLOW 999 LINES
17F1 21 FA 17         1090 ABORT  LXI H,MSG
17F4 CD 2A 02         1100        CALL SCRN
17F7 C3 16 00         1110        JMP RDY
17FA                  1120 ;
17FA 41 42            1130 MSG    DW 'BA'
17FC 4F 52            1140        DW 'RO'
17FE 54               1150        DB 'T'
17FF 0D               1160        DB 0DH
1800                  1170 ;
1800                  1180 ;TEST FOR CURRENT FILE
1800 2A 58 0D         1190 CFTST  LHLD BOFP
1803 7C               1200        MOV A,H
1804 B5               1210        ORA L
1805 CA 60 04         1220        JZ NFERR
1808 C9               1230        RET
1809                  1240 ;
1809                  1250 ;EQUATES FOR CASSETTE VERS 1.0 OF SOFT1
1809                  1260 RDY    EQU 0016H       ;RE-ENTER SOFT1
1809                  1270 SCRN   EQU 022AH       ;STRING PRINTER
1809                  1280 NFERR  EQU 0460H       ;NO CURRENT FILE ERROR
1809                  1290 ;
1809                  1300 ;STORAGE LOCS ARE IN ASCII BUFFER AREA
1809                  1310 LNHI   EQU 0DBDH
1809                  1320 LNLO   EQU 0DBFH
1809                  1330 ;
1809                  1340 ;FILE PARAMETERS
1809                  1350 FILE0  EQU 0D53H       ;CURRENT FILE NAME
1809                  1360 BOFP   EQU FILE0+5     ;POINTER TO START
1809                  1370 MAXL   EQU FILE0+9     ;MAX LINE NO.
```

WRITE AND IMAGE

Allen T. Fincher
Suffolk, VA

Simple stated, WRITE performs the opposite of READ (see Bill Blomgren's letter on page 16 of Vol. 3, #3). WRITE puts the contents of the specified area of memory into the given file verbatim; that is, without any additional bytes added to the written data.

When a file is written using IMAGE, the first two bytes written into the file are the length of the segment, the next two bytes are the load address of the segment, and then the data within the segment is written. If a start address is included, it is written last. This is the information printed on the console when the 'EXTRACT filename' command is used.

A program cannot be written on a disk and later executed by just typing its file name if it was put on disk using WRITE, because there would be no segment length, load address, or execution address included in the file for PTDOS to use. I am not referring to a program that was read off a disk using READ, but instead, to a new file.

Also, an existing image file cannot be put into memory using READ and then written to disk using IMAGE and have it run when subsequently loaded by typing its file name. Instead, it must be written out to disk using WRITE to that the proper segment length and load address which are also in memory along with the program code will be the only ones in the file.

Note the example of the IMAGE command at the bottom of page 2-25 of the PTDOS manual. It shows a file that will have several areas of memory contained within that file, all of which will be loaded with their respective data or code when the file is reloaded by typing its name. This can be done because the load length and load address of each segment is recorded in the file along with the data for the segments when IMAGEd.

LETTERS TO THE EDITOR:

..BELLS ON MY SOL

I have always envied terminals with an audible BELL which responds to control-G. Now my SOL has the makings of that feature.

Radio Shack's solid state buzzer 273-060 sells for $2.99 and will operate directly from the SOL parallel port! The loudness is about right too. Just connect the black lead to pin 1 or 2 of an appropriate connector and connect the red lead to a data output bit (pins 18-25). The buzzer can be turned on by:

```
     MVI  A,0FFh     ;turns on all 8 bits
     OUT  0FDh
```

and can be turned off by:

```
     MVI  A,0        ;turns off all 8 bits
     OUT  0FDh
```

Of course you'll need more sophisticated code if you are using other parallel output bits for other things.

Now when remote users of my system want to get my attention they run a CP/M program called BEEP. I still haven't figured out a useful way to use control-G because CP/M converts it to two letters (↑G) before it echos it to my CP/M console output routine.

Also, here are my dues for 1981. Keep up the good work.

Sincerely,
Dick Greenlaw 2/1/81
Dick Greenlaw

..PRODUCT REVIEW

18 March 1981

Since I find product reviews one of the most useful features in Proteus/News here are three brief ones that reflect my experience over the past two years.

1. Paper Tiger Printer. I replaced my old TTY with an IDS 440 and after eighteen months of moderately heavy use it continues to perform flawlessly. It is connected to the parallel port so it prints at maximum speed but can be used with the serial port if preferred. The chief reason I selected an IDS unit was to get the graphics option since I do lots of plotting. The technique is quite clever; the seven dots of the print head are arranged in a vertical column and the binary bit configuration of ones and zeros controls which pin head needles are fired. Programming graphics is like using machine language so it's tedious but the results are accurate and repeatable.

Incidently, I donated the ASR-33 to a deaf group; they have a great need for TTY's in working condition.

2. Central Data Dynamic RAM. Another gqod product; install and forget. I wanted to add the top 32K to my system to accomodate ALS8. The excellent CD manual uses as an example,complete with diagrams, the Sol with SOLOS occupying the C000 block to demonstrate their deselect feature using mini-jumps so even a duffer like me couldn't get it wrong.

3. Exatron Stringy Floppy. This time a real fiasco. Their literature looked good. I was espcially pleased to find that they used Proc.Tech. ECBasic with modified I/O. I won't go into the gory details but delivery promises were broken time after time and when the unit was finally shipped, it didn't work. EXATRON's suggestion was that I replace all 64K of dynamic memory with static memory. That didn't sound very cost-effective and I declined. Not recommended.

Len Kalish
580 S.San Vicente Blvd.#3
Los Angeles,CA 90048

..QUESTIONS ON SOL MODS, PRODUCT REVIEW

I have several questions for you or other PROTEUS members. First, has anyone upgraded their SOLs to the 4 Mhz 8080 CPU and if so, how was that done. Second, I see on my SOL PC board a video expantion interface. What is that for? The only reference to the interface is in the parts list and in the X drawings. Third and last of the questions, how could I make the SOL jump on reset to E800 rather than to C000? It would be much more convenient for me to have my North Star boot up on reset.

I recently completed the NOISEMAKER II programmable sound generator kit from Ackerman Digital Systems. The two AY-3-8910 programmable sound generators each have 3 12-bit tone generators, 1 4-bit amplitude control for each of the 3 tone sources, 1 5-bit noise generator, and 1 16-bit envelope generator. Also, each AY-3-8910 has 2 I/O ports for user applications plus, the tone generators may be used as D/A convertors rather than for sound production. Overall, the kit was fairly simple to make, and to my surprise, it worked the first time. The variety of sounds that might be produced with the 6 tone generators is fantastic as is the quality of tone. The board is a standard S-100 and it has a breadboard area where addition of on-board memory could be accomplished. Except for the vague instructions on how to address the individual PSG (programmable sound generator) and the minimal number of examples of sound generation, I would have to give give this project an EXCELLANT rating. For more information pertaining to the PSG, I found the data manual for the chip to very informative. Also, the data manual gives many examples for sound generation and the more complex musical compositions. The tones may be produced between the low of 30.5 Hz to the high of 125 kHz with a 2 mHz input clock. My computer now plays music!!

Sincerely,

ANDREW R. BOND
Box 233
Graton, CA. 95444
(707) 823-1232

...ON HELIOS IN A Z-80 SYSTEM

A while ago, Joe Maguire was asking about running a Helios on a 4 MHz Z-80 system. I have been doing this for a while and I'll be glad to correspond with anyone having problems with such a combination.

Tom Quinn, Route 2, Box 234K, Eatonville, WA 98328.

..HELP NEEDED ON MICROPOLIS DISK 1053 11

Is there anyone out there or do you know how to put P.T.s' Game programs and Basic on my Micropolis Disk 1053 11 to operate under CPM. I'm a complete novice but do really enjoy learning about my computer,so if someone can help please make it with simple stupid instructions. Thank you.

Bruce G. Diller
18651 E. Gallarno Drive
Covina, CA. 91722
(213) 966-0710

...ON USING BOTH SIDES OF A FLOPPY DISKETTE 2/1/81

I am now using both sides (one at a time) of many of my diskettes even though I have single-sided drives. It certainly saves a lot of money and space.

No, I didn't fork over $12 for a punch. I was encouraged by reports it is easier than that, and it is. The following procedure is for 5" diskettes, but should be about the same for 8" diskettes.

To convert regular minidiskettes to flip-over dual-sided diskettes (for use on single sided drives) I use an ordinary paper punch which makes a hole about the size of the sector hole window already present. I got a punch at an Ace hardware for less than a dollar with a plastic crap catcher. It's the pliers type, but with a mild curve to the handle so I don't have to bend the diskette too much to get in through the center hole.

I protect the diskette surface by using a clean work surface and by inserting half an index card into the center hole so it is always between the punch and the magnetic media.

Procedure: make a template using carbon paper if necessary. You want a sturdy but thin square outline of a diskette envelope with the sector hole, guide notches and write protect notch carefully marked. The center hole is not needed. Punch and cut out the two holes. The guide marks are just to help distinguish the two sides.

Allign the template on the back side of a diskette with the guide notch marks on the marked (read hole) edge of the diskette. Mark the new sector hole and the write protect notch. A red ball point pen will work. Turn the diskette with template over and put the template on top with the same orientation. Mark the new sector hole on this side.

Put the protective paper or card into the center hole on either side, centered on the new sector hole location. Bend the diskette carefully to allow the punch to get to the new sector hole location on that side via the center hole. There is a stretchy plastic liner as well as the cardboard. They are not fused together, so be sure to get both in the punch jaws. The plastic prefers to stretch, so punch several times in the same place and turn the punch while it is closed to get good cutting action without loosing allignment.

Then do the same on the other side and punch the write protect hole on the edge. A semi-circle worked ok with my Micropolis drives. The punch is slightly wider than Memorex's square holes. That is great, because my drive sometimes misreads the Memorex hole with disastrous results if I don't notice the l.e.d.

Format while you are doing the next diskette.

I see only one disadvantage: you could want files from both sides at the same time! Because of that, I am using this technique mostly for archival storage and inactive diskettes.

Dick Greenlaw
251 Colony Ct.
Gahanna, Ohio 43230

EVERYTHING YOU EVER WANTED TO KNOW ABOUT THE USE OF THE H-L REGISTER PAIR IN THE SOLOS/CUTER EXECUTION OPERATION BUT DIDN'T KNOW YOU WANTED TO KNOW

In Michael McKelvey's article in Vol. 3, #4 about transferring CUTS cassette files to PTDOS image files brought up a question I've seen raised in past issues but about which little has been explained. The reason the H-L register pair setting is important is that when the EXEC command is given in SOLOS/CUTER, the H-L register pair is loaded with the starting address of SOLOS or CUTER, whichever is being used. So in SOLOS, H-L will contain C000H when leaving SOLOS after performing the EXEC command.

The reason for this is mainly for running programs using CUTER. Because CUTER can run just about anywhere in the computer's ram area, programs written would not know where the CUTER I/O vectors were unless that program's I/O calls were changed. So what is done in SOLOS/CUTER compatible programs is that when first entered, the H-L register pair is saved in memory. Then when an I/O operation is to be performed, the following routine would find the proper SOLOS/CUTER vector location.
Let's say that a keyboard read from SINP is to be performed.
Then a CALL to the following will find where to go:

```
GETKEY: LHLD STARTADDR  *SOLOS/CUTER start addr saved on entry
        PUSH D          *Preserve D-E
        LXI  D,19H      *Offset to SINP
        DAD  D          *Make H-L point to SINP
        POP  D          *Restore D-E
        PCHL            *Jump to SINP, use its return
```

Or the actual call operands within the program may be over-written during initialization to point to the various I/O vectors as follows:

```
        LDHL STARTADDR  *Was stored on entry to program
        LXI  D,19H      *Offset to SINP
        DAD  D
        SHLD GETCH+1    *Over-write CALL addr value
        .
        .
        .
        .
GETCH:  CALL 0000H
```

I hope this helps clear the confusion over the H-L register problem.

Allen T. Fincher
Suffolk, VA

..TARBELL ON SOL

...If anyone in the area is interested, I'm adding modified Tarbell single density controller boards to SOLS. After they are added, the SOL will run PTDOS or CP/M merely by pressing the right key on boot. It makes the SOL more versatile. Also will have a SOL 2E parallel printer driver running under CP/M soon. My home phone is now (213) 345-3662.

Regards,

Jack Kinney

UNCLASSIFIED ADS

Do YOU have any SOL/HELIOS programs which the other members of Proteus could benefit from? Help others from having to re-invent the wheel! Send your disks to Chuck Athey, Helios Disk Librarian
5571 Shorehaven Circle
Livermore, Ca 94550

To those who donated programs for the latest H-7 library disk my thanks. If I inadvertently messed your programs up please let me know, keep them coming!
Chuck Athey (415) 449-8337

---

1 Diablo 1620 KSR for sale, very good condition. $1400 or best offer. Call Chuck Athey @ 415-449-8337 or 5571 Shorehaven Circle, Livermore, Ca 94550

For sale: FMT and MACRO as described in 'Software Tools', both with major inhancements, ie. Table of Contents generation in FMT..... Either will compile using the small C compiler by Ron Cain or BDS-C. 8080 Assembly version also available. Either for $30.00 on Helios disk or $25.00 on Cuts Tape. CP/M IBM formatted disk also available. $10.00 discount for both. Contact Chuck Athey Sunrise Computer, (415) 449-8337, 5571 Shorehaven Circle, Livermore, Ca 94550.

FOR SALE: SOL 20 Rev. E . Excellent condition with North Star Dick Controller and Two SA-400 drives and assorted games and software (including CPM and Whatsit). $1800. or best offer.
Mike Erickson, Alphanetics Engineering Consulting
P.O. Box 597
Forestville, CA 95436 (707)887-7237

FOR SALE: SOL 20 Less keyboard - no extra memory, but otherwise complete. $300.00
Dennis Polito
2411 Lincoln Ave.
Belmont, CA 94002 (415)592-5319 or 595-3949(answer. machine)

FOR SALE: Helios II controller and formater boards, $300.00 or best offer. (Drive not inluded - use as backup boards). PTC 16KRA memory board $100.00 32K static Dytron memory board (See Solus News June 1978) $300.00.
Ron Parsons
9001 Laurel Grove Dr,
Austin, TX 78758 (512)836-2514

COMPLETE SOL WITH NORTHSTAR FOR SALE: Sol, 32K Northstar minifloppy, Integral Data Systems printer. $1200 or best offer.
BobTyler
2329 Thompson Pl.
Santa Clara, CA 95050 (408)244-4457

Feb. 9, 1981

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

Dear Stan,

Attached is an article for PROTEUS. This one deals with a set of assembly language subroutines for use with MICROPOLIS BASIC. Some were designed when I converted several ECBASIC programs from the Proteus Cassette Library. The current set was done after a request from Paul Beauvais of Oxnard for a way to provide cursor controls which would facilitate his transfer of a business program from ECBASIC to MICROPOLIS BASIC. My thanks to him for the inspiration and the push. Of course, a copy will go to Lewis for inclusion in the Cassette Library.

Most recently my efforts have been bent towards moving (and enhancing) SOLOS to higher memory. I choose to add six new commands:

1. FILL addr1 to addr2 with hex byte.
2. MOVE from addr1 through addr2 to addr3 and up.
3. COMPARE and display mismatches from addr1 through addr2 with data at addr3 and up.
4. CS is cold start (MICROPOLIS cold boot at F400H).
5. CW is CP/M warm start at 0H.
6. MW is MICROPOLIS warm start at 4E7H.

I also put in a routine to initialize my CROMEMCO TU-ART on SOL power up (not strickly needed, but good practice).

The expanded SOLOS is now located at E000H in a pair of 2708's. The overflow (about 150 bytes) is in another 2708 at F000H and is located on a SOLID STATE MUSIC SYSTEMS PB-1. I decided on E000H for the new SOLOS instead of the more common F000H since it saved moving the MICROPOLIS controller from F400H. The only change to the SOL-20 was to bend U22-9 out so it did not enter the socket and then solder a jumper from U22-9 & U22-12.

The addition of a GODBOUT RAM-XX 24K static memory to go with my 32K DYNABYTE, gives me 56K of RAM plus SOL RAM & room for about 3.5K of EPROM (as yet unassigned). Now my Digital Research PL/I-80 has room to compile and link at least 30K source files.

I had just finished these changes when my copy of PROTEUS Vol. 3 No.5/6 arrived with Bob Stek's letter in it. I found the same problem with ERAS1 (although in my case I could fix it by changing the CPI 0D0H to CPI 0F0H). However, Bob didn't mention one other problem, namely, a typo in the CP/M source file. The error is in the GTBYT routine and prevents correct byte-mode cassette operation. The second MOV M,A should be a MOV A,M.

Note that four bytes of code from C037 to C03A are marked as part of the start-up routine. For the life of me, I can't find a need for it in SOLOS. Maybe it is needed for CUTER compatibility. I have left it out of my extended version of SOLOS with no ill effects yet!!

Yours truly,

MELVIN M. DALTON
7826 WEST 80TH STREET
PLAYA DEL REY, CA 90291

35/each

TABLE OF CONTENTS

NEW PHONE NUMBER FOR PROTEUS

Write it down. (415) 368-2300. That's your direct line to Proteus. We were using my dental office phone before, but the volume of calls was too much to bear. So please don't use any other number for calling us, but (415) 368-2300. Jane, our executive secretary, will answer the phone every weekday. If there is no answer, we are out. This way you won't have to pay for a long distance call, only to get my answering service. The best time to call to speak to me is Wednesday [redacted] P.M Pacific time.

--Stan.

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, telephone (415) 368-2300.

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.

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

Joe Maguire
PO Box 3742 DT
Anchorage, AK

99510

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

May, June
July, August 1981

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

Vol. 4 #3/4
Single Issue... $4.(US)

Safety Hazard Alert:

THERE'S A DISASTER LURKING IN YOUR HELIOS!

by Stan Sokolow

It was just sitting there, idle, with the power on and diskettes spinning, when a wisp of smoke emerged from the slots of my Helios. Rapidly the smoking increased as I fumbled for the power key and shut it off. Smoke continued to escape for 5 or 10 more seconds and the room filled with the pungent odor of burnt plastic. Something in the back of my mind flashed onto a conversation about the "mandatory" field-service notice that Processor Tech issued just before going out of business. Could it be that I was victim of that capacitor problem?

Before the smoke had faded away, I ran and grabbed a screwdriver, removed the three screws holding the Helios lid, fumbled to get it off, and examined the guts inside. Disk drive...looks normal. The power supply...looks normal. No, wait, the small aluminum can capacitor has a white wisp of smoke under it. Carefully scanning the bottom of the can where the terminals are and smelling around the cabinet convinced me. The capacitor had cooked itself. I just hoped it was not smoldering inside....

This is not just drama. It really happened a few days ago. When I had the time, I disassembled the Helios back panel and removed the regulator PC board from the base. The underside showed a definite charring between the terminals of the capacitor I suspected. Removing the screw terminals, I lifted the capacitor and discovered the PC board was burnt to a charred crisp in a half-inch diameter circle between the two terminals, and the end of the capacitor was covered with a black goo (which for all I know may be carcinogenic PCB's).

I have dramatized this story, because I suspect that most Helios owners still have the same problem lurking in their units. The cause of the problem is that the thinner of the two aluminum can capacitors in the Helios is a 40 working volt electrolytic, but 50 volt is required for reliable operation. Processor Technology discovered this soon before they shut down and they wrote an engineering change order and update memo, which I discovered in the junk discarded. I published it on page 22 of Proteus News, vol. 2, no. 4, July/August 1979.

But I didn't bother to make the change myself. I was just too lazy. My Helios worked okay, and the retro-fit notice didn't explain the consequences of leaving the status quo. It did imply, however, that PTC was scared about the capacitor's safety. Rightly so. If I hadn't been there at the moment of the failure and acted so quickly, the damage could have been extensive, and conceivably a fire could have started in the Helios cabinet. The Helios could have been a total loss in a matter of minutes.

At the time of this writing, I am waiting for an order of 50 volt 7700 mfd capacitors so I can get the Helios back into operation and see if the rest of the electronics is okay. I

DON'T MOVE SOL TO F000
MAKE SOL VANISH!

by Stan Sokolow

We've published several articles by members and by vendors about ways to get more contiguous memory space in your Sol so that you have more room for large programs. In this issue, we have two articles about modifying the Sol to make its built-in ROM, RAM, and video display respond to addresses in the F000 area of memory space, instead of the C000 space where they standardly respond. This of course would allow you to install RAM boards that provide memory from 0000 through EFFF addresses, for 60K of contiguous space.

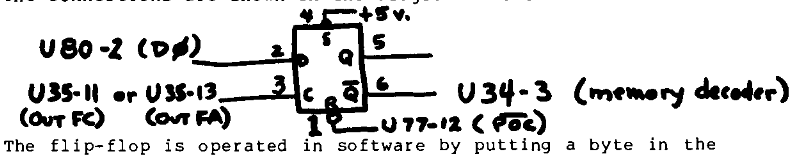
But, WAIT A MINUTE! Here we have a way to get a FULL 64K of RAM, and still be able to run your software that needs Sol at C000. You don't have to cut anything on your Sol to do it. And, best yet, it only costs peanuts.

Credit for this goes to Jim Spann for the idea and to Chuck Athey for modifying the idea and sending it to Proteus. Under the control of software, your program can disable the Sol's address decoder that identifies the addresses in the Sol's C000 through CFFF space. The Sol will then not respond to these addresses until the decoder is re-enabled by your program. When the Sol is "enabled" and does recognize an address in its space, it automatically disables the other memory boards in the system by pulling down the Phantom line on the S-100 bus.

Thus, the Sol can exist as another "bank" of memory, overlapping your memory boards, and not conflicting with the other memory. When your program wants to write onto the video display, or use Solos, or use the Sol's built-in RAM, your program first re-enables the Sol's decoder, accesses the Sol space, and again disables the Sol decoder. In a CP/M system, this would naturally be done by the routines in the CBIOS that do console I/O.

This is essentially a bank-selected memory scheme for Sol, but it doesn't conflict with other bank-select schemes that are used in some memory boards.

Here's how it's done. One extra IC is needed, a 74LS74 integrated circuit. This is a flip-flop that can be set/reset. The connections are shown in the diagram below.



The flip-flop is operated in software by putting a byte in the accumulator (register A) of the processor, with the lower order bit (bit 0) set to a 1 to make Sol go away, and a 0 to make Sol come back as normal. Then this byte is output to port FC, which is unused but decoded in the Sol. (It was reserved by PTC for an "alarm" feature.) The Power On Clear signal resets the

CON'T FROM PAGE 1: SAFETY HAZARD ALERT

just hope the smoke didn't do any damage to the optical scales used by the drive to tell where it is and to generate timing signals.

I have several Helios units and am avoiding their use as much as possible until I replace the capacitor in all of them. The capacitor was not easy to find, even here in Silicon Valley, so I have ordered a few extra ones to resell to any Proteus members who need to replace theirs. I will order more if the demand is sufficient.

To make it easy for you, I will put together a kit containing the components and illustrated, step-by-step instructions, so that anyone can make the change. No soldering is required, and the only tools are screwdrivers you can find at any hardware store. Also included will be fuses of a smaller value, to provide better protection as recommended by the retrofit notice.

The instructions will also repeat Helios Change Notice #3 which brings the regulator board from revision B to revision C. It should be done, too, and since the board will be out of the box anyway, now is the time. It involves a possible short circuit of the main power regulator.

The way to tell if your Helios needs the capacitor changed is to look inside the cabinet at the upright aluminum can on the right rear as seen from the front. Look for "40 WV" on the can. If you find it, you need the retrofit kit. If you find "50 WV" someone already made the change. Your fuse holders should also have been relabeled with smaller values.

Price of the Helios Safety Retrofit Kit is $35, plus $3 shipping & handling. (Please, no purchase orders. US funds only.) California residents add appropriate sales tax. Please allow 4 weeks for delivery, since the capacitors are hard to get if we run out of the stock on hand.

CON'T FROM PAGE 1: DON'T MOVE SOL TO F000 - MAKE SOL VANISH

flip-flop for normal Sol operation, so that the automatic jump to Solos on power on or system reset is not affected.

Not shown on the diagram are the power and ground pin connections, which could be obtained by "piggy-backing" the 74LS74 onto another 14 pin IC and soldering pin 7 to pin 7 (ground) and 14 to 14 (+5v) with the other pins bent out. The Sol manual appendix V show the pinouts of the IC's used in the Sol, including the 7474 IC.

The beauty of this modification is that Solos can stay where it is and be used by your existing programs. New programs can take advantage of the extra memory space and only return to the old configuration when needed. The content of the RAM that overlaps Sol is not affected and will return to normal when the Sol is disabled, provided that the Sol is jumpered for use of the Phantom signal (Phantom jumper installed).

The output port and bit position can be any other one that Sol decodes and which is not used for some other purpose. In another article in Proteus News, we will make a table of all of the I/O ports and their assigned functions, so that as modifications are thought up, we can keep some standardization to the new Sol features. Actually, I feel that the FC port should be reserved for some use that needs a full 8 bits.

My inclination is to modify Chuck's circuit to use bit 0 of output port FA, which is also unused and the other bits of the port already have control functions. To do this, connect pin 3 of Chuck's flip-flop to pin 13 of U35 instead of pin 11 of U35.

ENCYCLOPEDIA PROCESSOR TECHNICA
A COMPLETE REFERENCE

I've mentioned it before, and it's finally coming tegether. We are putting finishing touches on the first few volumes of Proteus's complete guide to the entire PTC product line. Here's an idea of what we will cover.

Volume 1 will be a technical manual on the Sol computer. It will incorporate material from the big black looseleaf notebook that came with early Sol's, plus material taken from the PTC Sol Service Manual, from Access (PTC's newsletter to users), from Proteus, and from other sources. We will cover topics such as: theory of operation, schematics, assembly drawings, parts lists, troubleshooting, testing, updating (rev D to rev R), constructing and using the ParaSol Debugger, input/output hookups, timing diagrams, testpoint voltages, hardware interfacing, etc.

Volume 2 will cover introduction and basic operation of the Sol, including the semi-technical Users Manual that came with later Sols. Topics include: functions of the Sol's buttons and switches, hookups to common printers, cassette operations, use as a dumb terminal, minor troubleshooting.

Also included will be a chapter on the history of Processor Technology Corporation and of the Sol computer. This is an area which I feel the microcomputer revolution is overlooking. We will include reprints of magazine articles which describe the way the Sol came to be, copies of old advertising, dealer lists, historical accounts from insiders, etc. Someday that walnut-sided box will look as strange to us as a Ford Model T or an old "woodie" station wagon. I'd like us all to have a book on the shelf that re-counts how the West was won, so to speak.

Volume 3 will be on programming the Sol. We'll cover: Solos, the Sol's BASIC languages, the 8080 machine architecture from the programmer's view, the 8080 instruction set, assembly language programming, communications interface programs, the Sol's system architecture (I/O ports, memory map, etc).

Volume 4 will be devoted to ALS-8. Included will be the user's manual, the users-group newsletters describing the internal routines, the source listing, etc.

Volume 5 will be on PTC memory boards, covering the entire product line, from the old 2KRO ROM board to the 64KRA-1 RAM board. We'll include schematics, assembly drawings, theory of operation, switch settings, engineering changes, upgrades, troubleshooting, ROM listings, parts lists, etc.

Volume 6 will cover PTC input/output interfaces, such as the 3P+S, the CUTS board, the VDM, and Hytype and SolPrinter interfaces.

Volume 7 will be on the Helios disk system, including revisions, testing, troubleshooting, theory of operation, etc.

Volume 8 will cover programming with the Helios disk system, including PTDOS, Disk BASIC, etc.

Volume 9 will be on using the Sol with other manufacturers' equipment, such interfacing to other disks, modifying memory boards to work properly in Sol, customizations for the Sol with CP/M, NorthStar, Micropolis, etc.

As you can see, this will be a comprehensive reference. That's why the name sounds like the "Encyclopedia Britannica." As time goes on, we will issue supplementary pages to insert into the volumes when we have new updates, etc. This will be a living resource.

We'll announce the volumes and prices in more detail when they are ready for sale.

Repairing Your Sol
By Joe Maguire

The Memory Boards

I had intended to begin this final segment with some general suggestions on trouble shooting accessory boards plugged into the Sol backplane but I have received so many requests for help on the PTC 16KRA memory boards That I'll get right to them.

First, I will be talking about the original 16KRA memory (and it's cousin the 32KRA) not the "N" KRA (designated the 16KRA-1) which was designed later.

Compared to all the excellent products which PTC produced, the 16KRA had to be the worst. It was designed in 1976 and, as far as I have been able to determine, it was the first dynamic memory on the S-1ØØ bus which used multiplexed addressing for the chips. The design of such an array is formidable, even today, and considering that PTC was leading the state of technology it's no wonder they had problems.

The requirements for the design were: to be able to use it on a variety of machines and to support the DMA used by PTC's Helios disk system. It almost made it. The major problem was timing. PTC found, after virtually every board they sent out was returned, that aging of the chips was enough to upset the critical timing. In desperation, PTC bought two German made 1ØØ mHz 'scopes just for trouble shooting. During a visit I made to the factory in 1978, (with a number of duds in hand) I was told not to attempt to replace any chips myself, that PTC individually selected every IC that went into every socket.

Well, faced with that kind of situation, what can you do? Unfortunately, not much. I'll give you some tips I've picked up here and there but if your 16KRA is working OK consider yourself lucky.

As an interesting sidelight, when I was attempting to research the 16KRA problems, no former PTC employee I contacted would admit to any knowledge of them. Only after considerable pleading did one former engineer tell me that he would give me the name of a person to contact but that "if you ever tell him that it was me that gave you his name, I'll never help you again!" Faced with that kind of warning I decided that discretion was the better part of valor. Success has a thousand fathers but failure is an orphan!

PTC did devise several fixes which seemed to help. Some of them were:

1. Install extra grounds on the solder side of the board by running #2Ø wire jumpers between ground traces.
2. Connect pin 53 of the Sol backplane to ground (pins 5Ø & 1ØØ) as per the new IEEE S-1ØØ standard.
3. Jumper the ground trace on the 16KRA to pin 53 on the board.
4. Remove the header and plug at P2 and solder jumpers directly into the holes on the board.
5. Solder the delay line, U71, directly to the board.
6. Replace the TI S-1ØØ connectors on the backplane with AMD type.

The TI (Texas Instruments) connectors are notorious for poor contacts. In the last run of Sols before PTC went under, the TIs were replaced with AMDs. You should not attempt to replace yours unless you have access to vacuum desoldering equipment otherwise you could ruin the backplane board.

A common problem with the 16KRA is the delay line. Due to the high current passing through it, it eventually fails. A failure can be recognized by loss of refresh or that the computer just hangs up when the CPU requests a memory access to the offending board. I have not been able to find any replacement for the delay line. It was a custom part manufactured for PTC and all specifications have been lost. The one used in the "N" KRA series will not work.

Some of the idiosyncrasies of the 16KRA are almost too weird to believe. Some users who own several Sols tell me that some boards will work in one machine but not the other. In one case, after working in Sol #1 for two years it ceased. Then it would only work in Sol #2 where it originally would not!

The bottom line comes down to this. Use your 16KRA (or 32KRA) for as long as it works. If it fails, try a few of the things I've mentioned, even swapping it with a friend. If nothing gets it going again, don't spend any money on it but get a newer type. Dynamic memories are now cheap and reliable. I have used a number of NorthStar dynamic memory boards for years without a single failure. The poor heat disssipation of the Sol makes the use of static memories risky unless increased ventilation is provided.

A few hints for accessory boards in general: Be sure to reseat them in the backplane slots from time to time. Don't use rubbing alcohol to clean the connector strip. It contains lanolin which can remain behind to cause poor contacts. A good grade of vodka would be better but I can think of a better use for that while trying to fix a cantankerous 16KRA! An ordinary pencil eraser does a good job of cleaning contacts. Do add the ground to pin 53 on all boards. That pin (sense switch disable) is vacant in the Sol and connecting it to ground can dramatically reduce noise.

Your Sol is a well designed piece of equipment. It can hold its own against any of those new "consumer toys." Remember, there are more 8Ø8Ø microprocessors in use than all other types combined. The government alone must be using millions of them. The 8Ø8Ø and its vast repertoire of software will be around for the next twenty years. With just a little care now and then, your Sol will be too.

Joe Maguire

```
    1 REM.   **********************************************
    2 REM.   *  Inputs and formats date                   *
    3 REM.   *  Program name       :     DATEIN           *
    4 REM.   *  Created            :     1Ø Jan 1981      *
    5 REM.   *  Version            :     1.Ø2             *
    6 REM.   *                     :                      *
    7 REM.   *  Checked            :     David L. Dalva II *
    8 REM.   *  Ref.               :     none             *
    9 REM.   *  orginal  program                          *
   1Ø REM.   **********************************************
   2Ø REM
1ØØØØ REM.       'DateIn'  Subroutine.   Use  APPEND
1ØØØ1 REM
1ØØ1Ø PRINT "\K"
1ØØ2Ø CURSOR 5,Ø: PRINT "Date  : [mmddyy]": CURSOR 5,9: INPUT (2,Ø)"",M
1ØØ3Ø IF M<1 THEN 1ØØ2Ø
1ØØ4Ø IF M>12 THEN 1ØØ2Ø
1ØØ5Ø LET M1$=STR(M): IF LEN(M1$)<2 THEN  LET M1$="Ø"+M1$
1ØØ6Ø LET A1$=M1$+"/": CURSOR 5,9: PRINT A1$;"ddyy]": CURSOR 5,12: INPUT (2,Ø)"",D
1ØØ7Ø IF D<1 THEN 1ØØ6Ø
1ØØ8Ø IF D>31 THEN 1ØØ6Ø
1ØØ9Ø LET D$=STR(D): IF LEN(D$)<2 THEN  LET D$="Ø"+D$
1Ø1ØØ LET A1$=A1$+D$+"/"
1Ø11Ø CURSOR 5,9: PRINT A1$;"yy]": CURSOR 5,15: INPUT (2,Ø)"",Y
1Ø12Ø IF Y<1Ø THEN 1Ø11Ø
1Ø13Ø LET Y$=STR(Y): LET A1$=A1$+Y$: CURSOR 5,9: PRINT A1$;"]": LET A$=A1$
1Ø14Ø PRINT "Date 'mm/dd/yy':  ";A$
```

***Dead Keys? on your SOL?? Shoot it, but!!!

Before you do, think--Hum---why bother Keytronics. Get a one hole paper punch, then some aluminum foil and an appropriate piece of foam. Punch out an aluminum foil disk, then punch out a column of foam--- got that ? then take some contact cement and glue your aluminum disk to your foam column. Take the whole thing and glue it into position where the faulty one was, works great saves money, time and nerves.

Bruce G. Diller
18651 East Gallarno Dr.
Covina, CA 91722

CPMUG ON NORTH STAR DD
Gib Zeratsky

6/23/81

813 Inlet Road
Green Lake, WI 54941

Subj: CPMUG Vols. On North Star DD

Dear Stan,

As I reported to you some time ago I have purchased the CP/M Users Group disks that I thought other members would be interested in. I got rid of all the programs that required a Z80 CPU and consolidated things down to five (5) North Star DD diskettes as follows:

CPMUG Vol. 1
CPMUG Vol. 11
CPMUG Vol. 14 & 15
CPMUG Vol. 29 & 42
CPMUG Vol. 36

I admit that my judgement as to what should be left off these disks may be questioned but none-the-less I did it. Any member who would like copies of any or all of these disks can order same from me for $8.00 per disk.

I am attaching a directory listing of these disks.

Regards,

Gib Zeratsky

THIS DISK IS AN EDITED VERSION OF CPMUG VOL.1 ITEMS FROM THE ORIGINAL VOLUME NOT HAVING USE ON SOL OR ON AN 8080 MACHINE HAVE BEEN DELETED. THE ORIGINAL IS ON TWO NORTH STAR DD DISKETTES.

GIB ZERATSKY

PRNT - IS A PRINT UTILITY SOMETHING LIKE TYPE BUT WITH PAGINATION AND HEADINGS. IT SHOULD RUN ON ANY CONFIGURATION OF CP/M WITH AN I/O DEVICE THAT RECOGNIZES A FORM FEED CHARACTER (HEX OC) AS A TOP-OF PAGE COMMAND. THE SYNTAX FOR USING PRNT IS:

```
PRNT FILENAME.EXT
```

PRNT ALSO MAY BE USEFUL TO THOSE WHO NEED AN EXAMPLE OF CP/M DISK I/O ACCESS.

PRINT - IS A PATCHED VERSION BY ARG TO COPE WITH HARDWARE THAT DOES NOT RECOGNISE OCH

EXAM - IS A DISK SECTOR EXAMINATION UTILITY FOR A SYSTEM EQUIPPED WITH A VDM. ONCE RUNNING EXAM WILL ALLOW VISUAL EXAMINATION OF ANY SECTOR ON THE DISK (DRIVE 0 ONLY IN THIS VERSION). THE CURRENT VERSION WILL ONLY OPERATE IN A 16K SYSTEM BUT THAT IS EASILY FIXED. THE COMMANDS TO EXAM ARE:

```
+  =   READ NEXT SECTOR
-  =   READ PREVIOUS SECTOR
I  =   STEP HEAD IN (TOWARD TRACK 76) AND READ
O  =   STEP HEAD OUT (TOWARD TRACK 0) AND READ
```

DUE TO A MINOR BUG WHICH I HAVEN'T HAD THE TIME TO FIX, THE FIRST COMMAND TO EXAM SHOULD BE A +.

MEMTST - IS A MEMORY TEST FROM THE INTEL USERS LIBRARY. IT HAS INTERNAL I/O ROUTINES AND MUST BE MODIFIED TO SUIT THE USERS SYSTEM. IT IS A VERY GOOD TEST OF MEMORY WHICH WE HAVE BEEN USING EXTENSIVELY IN THE LONG ISLAND CLUB. THE PROGRAM WILL DISPLAY MEMTST 1.0 AT WHICH POINT THE USER SHOULD ENTER STARTING AND ENDING HEX ADDRESSES SEPERATED BY A COMMA. THE PROGRAM WILL THEN TYPE ADDR: REF: ACT: AND BEGIN TESTING. ANY ERRORS WILL CAUSE AN ADDRESS ,REFERENCE DATA, AND ACTUAL DATA TO BE PRINTED.

MAZE - IS A MAZE PROGRAM FROM THE INTEL USERS LIBRARY. IT SHOULD RUN ON ANY SYSTEM.

TRAIN - IS THE VDM TRAIN PROGRAM FROM THE PROCESSOR TECHNOLOGY NEWSLETTER. JUST TYPE

PROCESSOR TECHNOLOGY BASIC/5 (SOFTWARE #2) PATCHED FOR CP/M DISK SAVE AND LOAD AND FOR LPRINT AND LLIST

*"This package is being made available, free of restrictions and cost .... to the individual hobbiest, clubs, schools, colleges and universities for their own computer systems.

Any commercial use ...... is strictly forbidden without...permission of Processor Technology Corporation."

*NOTICE INCLUDED IN SOURCE LISTING BOOKLET. ALSO AVAILABLE FROM PROCESSOR TECHNOLOGY IS "5K BASIC MANUAL" AS PRIMER TO BASIC IN GENERAL AND BASIC/5 SYNTAX IN PARTICULAR. SEE THESE TWO BOOKLETS PLUS THE BASIC/5.DOC FOR FULL UNDERSTANDING. (NOT ANYMORE YOU WON'T SEE THESE BOOKLETS - BORROW ONE)
STARTREK.TBI PREVIOUSLY SEEN AS FILES IN VOLUME 2 . NOTE THAT THE STARTREK LISTING APPARENTLY ENDS IN MID-SUBROUTINE. THE PROGRAM APPEARS TO RUN, ALTHOUGH A REAL TREKKY WILL PROBABLY EXERCISE THE FATAL PORTION AND WARP HIMSELF OUT OF THE TPA. [I BET KLINGONS SPEAK IN ZILOG MNEMONICS]

TYPE B:CATALOG.14
VOLUME 14 & 15 ARE COMBINED ON THIS DISK. ITEMS FROM VOL 15 ARE LISTED SEPERATLEY UNDER CATALOG.15

VARIOUS CP/M UTILITIES

NUMBER	SIZE	NAME	COMMENTS
		CATALOG.14	CONTENTS OF CP/M VOLUME 14
14.2	6K	BCD-DRVR	DRIVER TO CONVERT ASCII TO THAT FUNNY REVERSE BCD USED BY IBM 2740 TERMINALS
14.6	3K	DUMP.COM	RUNNING VERSION OF DUMP BELOW, SUPPLIED AS DUMP.ASM CODED FOR TDL ASSEMBLER (8080 OK)
14.7	23K	DUMP.MAC	FANTASTIC DISK VIEWER PROGRAM. CAN ADDRESS FILES, CP/M GROUPS OR SECTORS DIRECTLY, USES STANDARD CONSOLE OUTPUT, AND DISPLAYS IN DDT DUMP FORM WITH HEX AND ASCII SIMULTANEOUSLY
14.9	2K	MOVE.ASM	A PIP TO TRANSFER FILES WITHOUT THE PROBLEM OF <CTL Z>'S IN FILES WITH NON-COM TYPE NAMES, SUCH AS BASIC-E/CBASIC INT FILES - SEE DLOAD.DOC
14.10	3K	PUT.ASM	USED TO LOAD A FILE AT ANY MEMORY ADDRESS, AND OPTIONALLY START TO RUN IT. USEFUL FOR POKING ODD DRIVERS AND MONITORS INTO MEMORY FOR THOSE WITH NO FRONT PANEL
14.18	7K	SEDY.ASM	DISK PEEKING PROGRAM.
14.19	2K	SEDY.COM	COMPILATION, AS 14.18 WRITTEN FOR TDL ASSEMBLER (8080 OK)

CON'T FROM PAGE 4 : - CPMUG VOLS. ON NORTH STAR DD

```
A>TYPE B:CATALOG.15
VOLUME 15

UTILITIES AND NON-BASIC GAMES

NUMBER  SIZE    NAME            COMMENTS

                CATALOG.15      CONTENTS OF CP/M GROUP VOL 15
15.1    2K      ALLOC.ASM       SOURCE OF 8.1 WITH IMPROVEMENTS TO
                                ALLOW "ALLOC B:" AND GIVE DISPLAY OF
                                ONLY THE USABLE PORTION OF DISKETTE
15.2    2K      BEASTIE.ASM     SOL VDM GAME FROM DR. DOBBS
15.3    8K      CHASE.ASM       SOL VDM GAME FROM DR. DOBBS
15.12   16K     PRINT3.ASM      FURTHER RE-WRITE OF 8.23 TO PERMIT COMMAND LINE
                                CONTROL OF TITLES, PAGINATION AND PRINT PITCH
15.21   4K      ROMP.ASM        TARBELL TAPE LOADER AND SAVER
15.22   46K     SOLOS.ASM       SOL OPERATING SYSTEM PUBLISHED IN
                                ACCESS.  COMMENTED AND INCLUDES NOTED
                                PATCH TO USE AS CP/M CONSOLE
15.23   2K      TARGET.ASM      SOL VDM GAME BY GEORGE W. ROMPOT FROM
                                MITS NEWSLETTER
15.24   6K      TLOAD.ASM       RE-WRITE OF 6.18 WITH DOCUMENTED
                                ENHANCEMENTS
15.25   5K      TSAVE.ASM       RE-WRITE OF 6.20 WITH DOCUMENTED
                                ENHANCEMENTS
```

```
TYPE B:CATALOG.29

ASSEMBLER GAMES AND UTILITIES

VOL29 IS ANOTHER TWO DISK VOLUME. SOME FORTRAN STUFF AND
   FLOATING POINT PACKAGE FOR BASIC/E ARE DELETED.

NUMBER  SIZE    NAME            COMMENTS

                CATALOG.29      CONTENTS OF CP/M GROUP VOLUME 29
                VOLUME29.DOC    COMMENTS ON PROGRAMS ON VOLUME 29
29.1    25K     TTT.ASM         3-D TIC TAC TOE GAME
29.2    9K      CHASE.ASM       VDM/SOL GAME.  15.3 WITH CP/M I/O.  SEE
                                VOLUME29.DOC.
29.3    3K      TARGET.ASM      VDM/SOL GAME.
29.4    8K      DEFLECT.ASM     VDM GAME
                                SILENCE"
29.7    12K     PIRANHA.ASM     VDM GAME
29.9    3K      ROBOT.ASM       VDM GAME
29.10   2K      ALLOC.ASM       DISK ALLOCATION DISPLAY.  SEE VOLUME29.DOC.
29.12   5K      SAP.ASM         SORTS AND PACKS DIRECTORY.  IMPROVED VERSION
                                OF 19.8.  SEE VOLUME29.DOC.
29.13   3K      NOTATE.ASM      ADDS COMMENTS TO ASM SOURCE.  ASSEMBLE USING MAC.
                                SEE VOLUME29.DOC.
29.14   1K      SYMSTACK.LIB    ASSEMBLER TIME PUSH/POP LIB FOR MAC.  SEE
                                VOLUME29.DOC.
29.15   2K      NCOMPARE.LIB    CORRECTED VERSIONS OF DIGITAL RESEARCH
                                DISTRIBUTION.  SEE VOLUME29.DOC.
29.16   11K     SEQIO.LIB       CORRECTED VERSION OF DIGITAL RESEARCH
                                DISTRIBUTION.  SEE VOLUME29.DOC.
29.17   1K      WHENS.LIB       MODIFICATION OF DIGITAL RESEARCH VERSIONS TO
                                ALLOW INFINITE NESTING.  SEE VOLUME29.DOC.
29.18   1K      DOWHILES.LIB    MODIFICATION OF DIGITAL RESEARCH VERSIONS TO
                                ALLOW INFINITE NESTING.  SEE VOLUME29.DOC.
29.19   2K      SELECTS.LIB     MODIFICATION OF DIGITAL RESEARCH VERSIONS TO
                                ALLOW INFINITE NESTING.  SEE VOLUME29.DOC.
29.31   5K      TIMESQ.ASM      SIGN GENERATOR FOR VDM.
29.33   1K      VDM.DOC         MODIFY ED TO PAGE 16 LINES INSTEAD OF 24.
```

```
A>TYPE B:CATALOG.42

A FEW OF THE ITEMS FROM VOL. 42

VOL42 IS ANOTHER TWO DISK VOLUME. THE BULK OF IT DOES NOT
   CONCERN SOL. I GOT IT FOR THE FILES THAT CONVERT ALS-8.

NUMBER  SIZE    NAME            COMMENTS

                CATALOG.42      CONTENTS OF CP/M GROUP VOL.42
42.7    3K      HOW2BS.DOC      PUT BS IN YOUR BIOS
42.9    17K     LIFE8.ASM       FAST,COMPREHESIVE LIFE...
42.10   3K      LIFE8.COM       ...PROGRAM FOR THE VDM
42.11   2K      LIFE8.DOC       DOC ON ABOVE
42.14   4K      PTSRCNVT.ASM    Proc. Tech. Pkg. 1 or ALS-8..
42.15   2k      PTSRCNVT.COM    ...to CP/M file conversion..
42.16   2K      PTSRCNVT.DOC    ...program.
42.17   13K     QUOTES.PRN      Words of wit and wisdom re-
                                related to computers...
```

```
  TYPE B:-CATALOG.036
VOLUME 36

Assemblers, editors, text processor related programs,
memory bank boot and manager, misc. utilities,

ML80 LANGUAGE AND TED (A LINE ORIENTED TEXT EDITOR)
HAVE BEEN DELETED FROM A TWO DISK VOLUME

NUMBER  SIZE    NAME            COMMENTS

                -CATALOG.036    CONTENTS OF CP/M VOL. 36
        6K      ABSTRACT.036    Abstract on some files
36.1    7K      COMBINE.ASM     Combine multiple .asm files..
36.2    1K      COMBINE.COM     ..and delete comments
36.3    4K      CPM-PEN.ASM     Convert CP/M to electric..
36.4    1K      CPM-PEN.COM     ..pencil files
36.11   6K      LINKASM.COM     ASM.COM with feature to link..
36.12   5K      LINKASM.DOC     ..files together at assembly
36.13   1K      LIST.COM        List files with starting..
36.14   1K      LIST.DOC        ..line number
36.20   3K      MFACCESS.LIB    Subroutine to setup for..
                                ..multiple files (ex: *.ASM)
36.23   3K      PEN-CPM.ASM     Electric Pencil to CP/M file..
36.24   1K      PEN-CPM.COM     ..conversion program
36.25   2K      PEN-CPM.DOC     ..
36.26   24K     POW.ASM         Source for text processor
36.27   5K      POWCMDS.POW     POW documentation
36.28   2K      POWTEXT.POW     More POW doc.
36.29   6K      SCRAMBLE.ASM    File encode/decode
36.30   2K      SCRAMBLE.DOC    Doc on above
36.33   10K     TOP.AZM         TDL Text output processor..
36.34   5K      TOP.PRN         ..patches
36.35   3K      TOP.TOP         ..
36.36   22K     XREF.ASM        Cross references .ASM files
```

Marcia Todd of SIMON PUBLIC RELATIONS, 1230 Oakmead Parkway, Sunnyvale called PROTEUS to tell us about the excellent repair job done on her North Star by MICRO COMPUTER BUSINESS SYSTEMS, Dennis Lewandowsky and Dennis Wong, 2094 B Walsh Ave., Santa Clara, CA 95051, (408-496-6227).

MODIFICATION OF CP/M CBIOS FOR AJ841 PRINTER
BY Deslar K. Patten

Bennetts Photo Service

March 19, 1981

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

Dear Stan:

I have included a source listing of a list driver for CPM using the AJ841 printer. The article by Lewis Moseley in the Proteus News was an autopage driver which has been modified for CPM.

This routine keeps track of the number of carraige returns sent to the printer and when a form feed character is sent, it will output the required number of carraige returns to get to the bottom of the page. Following are some implementation notes:

1. If no form feed characters are sent it will not page eject. Otherwise it conflicts with software that inserts its own form feeds.

2. When using PIP, the computer loops back through warm boot at the end of every command. This is useful for reseting the "LINES" counter for the next printing operation. Note the two lines, "MVI A,Ø" and "STA LINES," in the warm boot subroutine.

3. The AJ841 doesn't recognize ⟨LF's⟩ after ⟨CR's⟩, so line feeds are not counted. The only problem that could occur is if line feeds are sent without CR's. I haven't had that problem yet, however, a simple trap could be used to look for LF's without a preceding CR.

4. The printer is set to print at the tear line before each printing operation. The driver will then either advance to the top margin, or in some software with an initial page eject, will advance one page and then advance to the top line margin.

```
;CPM ENTERS HERE WITH CHARACTER IN REG C
;AJ841 TERMINAL IGNORES ANY LF AFTER A CR, BUT AUTOMATICALLY
;GIVES A CRLF FOR EVERY CR RECEIVED.
;
LIST:           PUSH            B     ;SAVE ORIGINAL CHAR.
                MOV             B,C   ;MOVE CHAR. TO B FOR OUTPUT
                MOV             A,C   ;MOVE CHAR. FOR COMPARE
                ANI             7FH   ;STRIP PARITY BIT
                CPI             FF    ;CHECK FOR FORM FEED
                JZ              FFEED
                CPI             CR    ;CHECK FOR CARRIAGE RETURN
                CZ              CRFEED
                MVI             A,2   ;SELECT PARALLEL PORT
                CALL            AOUT  ;OUTPUT IT THRU SOLOS
                POP             B     ;GET ORIG CHAR BACK
                MOV             A,C   ;PUT IN A FOR SOME PROGRAMS
                RET
;
;FOR EVERY CR RECEIVED THIS ROUTINE INCREMENTS "LINES" TO
;KEEP TRACK OF LOCATION ON THE PAGE.
;
CRFEED:         LDA             LINES ;GET # OF LINES SOFAR
                INR             A     ;ADD ANOTHER LINE TO COUNT
                STA             LINES ;STORE NEW COUNT
                RET
;
;THIS ROUTINE PUTS OUT ENOUGH CR'S TO FILL THE REST OF THE
;PAGE. IT THEN OUTPUTS SOME MORE TO GET TO THE FIRST LINE
;ON THE NEXT PAGE.
;




FFEED:          LDA             LINES ;GET # OF LINES SOFAR
                MOV             C,A
                MVI             A,PAGE  ;# OF LINES ON A PAGE
                SUB             C      ;A=LINES LEFT ON THIS PAGE
                JM              FFEED3  ;MUST BE NON STANDARD PAGE
                MOV             C,A    ;C=COUNTER FOR LINES LEFT
                CPI             0      ;CHECK IF PAGE IS FULL
                CNZ             FFEED1  ;OUTPUT CR'S UNTIL NO MORE LINES
FFEED3:         MVI             A,TMARG
                STA             LINES
                MVI             C,TMARG ;LINES NEEDED FOR TOP OF NEW PAGE
                CALL            FFEED1  ;OUTPUT CR'S FOR NEW PAGE
                POP             B      ;GET BACK ORIG CHAR
                MOV             A,C
                RET
;
;THIS ROUTINE OUTPUTS CR'S UNTIL COUNTER IS EMPTY (REG C)
;
FFEED1:         MVI             B,CR
FFEED2:         PUSH            B      ;SAVE COUNT & CR CHARACTER
                MVI             A,2    ;SELECT PARELLEL PORT
                CALL            AOUT   ;OUTPUT THRU SOLOS
                POP             B      ;GET BACK COUNT
                DCR             C      ;SUBTRACT ONE LINE
                JNZ             FFEED2  ;LOOP UNTIL COUNTER IS ZERO
                RET
;
LINES:          DS              1      ;STORAGE LOCATION FOR LINE COUNT
;
                END
```

CON'T FROM PAGE 6: MOD. OF CP/M CBIOS FOR AJ841 PRINTER

```
BOOT:           ;GETS CONTROL AFTER COLD OR WARM BOOT
        MVI     A,0
        STA     LINES
         .
         .
         .
        REMAINDER OF YOUR USUAL BOOTLOAD ROUTINE
         .
         .
         .
```

SOLOS MODIFICATION FOR CP/M
BY Deslar K. Patten

Bennetts Photo Service

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

Dear Stan:

Several issues back, Thomas McGee wrote an excellent article on output routines for the SOL. I was very surprised not to see any information about the carriage return problem and ways to solve it. This basically is an open letter to Thomas McGee with some additional information on output routines.

Processor Technology has included a "clear to end of line" function in the SOLOS module. (I don't know why and hopefully somebody who does can answer that question.) The "clear to end of line" function works with programs written for the SOL, but causes problems on some programs not written for it.

CPM as an example, sometimes will output a carriage return only, and then after some processing will output another carriage return and line feed. The second carriage return will "clear to end of line" and erase the previously entered line. This is particularly frustrating when an error has been made in entering a command, and CPM comes back with an "Invalid Command," but has erased the line in the meantime. I have worked with a number of terminals and so far have found the SOL to be unique with this problem.

There are three methods being used to correct the problem in CPM. If anyone knows of another, please let me know.

1. SWITCHING CR & LF - A routine that outputs a CR for an LF, and an LF for a CR. This works unless the software being used has definite ideas about which should be which. An example would be a screen oriented editor like Wordstar.

2. CHARACTER TRAP FOR 2nd CR - This routine stores the last character output and then compares the newly received character to it. If two CR's are sent, one after the other, the second CR is discarded or "trapped." I don't know how well this works.

3. CHARACTER TRAP & SWITCH - This routine combines both of the above. First, two sequential CR's are detected as in #2. Then the second CR is replaced with an LF. This generates extra line feeds not expected by some software such as Wordstar.

It was at this point that I came to the conclusion that the "clear to end of line" had to be eliminated. The only conflict that could exist is with Processor Technology software that may use that function. After making the change indicated below, I have run BASIC5, EXTENDED BASIC, and TREK-80 and haven't had any problems.

Software adapted for use on the SOL may use the CR as an "erase to end of line." In that case, the software would have to be changed. As an example, Wordstar, after installation, assumes a carriage return will "EOL." By referring to the manual, the "EOL" function can be eliminated easily by changing the specified location to zero. The built-in software then performs the function.

After receiving the SOLOS.ASM file from the CPMUG library, I have found that someone has already made this change. The approach was to XRA A to zero before calling PCUR instead of MVI A,64 as I have done. Both seem to work fine.

The CPMUG, SOLOS.ASM, however, didn't maintain the original address locations for subroutines. This can be corrected simply by adding NOP's to fill the address locations no longer used. For the carriage return modification, after XRA A, two NOP's should be inserted.

The other location where a problem occurs in the CPMUG version is in the PCLOS routine. The original Personality Module (2708) and the original SOLOS listing have two MOV A,M by mistake. The second MOV A,M was removed in the CPMUG version. All that is required is to insert an NOP after the MOV A,M.

This will change the CPMUG version to have the same addresses as the original for any programs making direct calls into SOLOS subroutines.

Two other changes have been made which work well for us and haven't created any problems. One was to make the SOL boot up in CPM upon reset or power on, and the second was to make "Mode Select" go to "Command" mode at any time. Since the "TERMINAL" mode was never used, the space was used for the modifications. Note, however, that the proper number of unused memory locations are filled with NOP's to maintain the same address locations as mentioned above.

Sincerely,

BENNETTS PHOTO SERVICE, INC.

Deslar K. Patten
R&D Project Engineer

Enclosure
DKP:sb

CON'T FROM PAGE 7 : SOLOS MODIFICATION FOR CP/M

```
12200   ;
12300   ;   KEYBOARD INPUT DRIVER
12310   ;
12312   ;       THIS ROUTINE HAS BEEN MODIFIED TO JMP TO "CPMCMD",
12314   ;WHICH THEN CHECKS FOR A "MODE SELECT". IF RECEIVED, THEN IT
12316   ;GOES TO SOLOS "COMMAND MODE", OTHERWISE IT WILL RETURN AS IN
12318   ;THE ORIGINAL VERSION. THIS ALLOWS ENTERING THE COMMAND MODE
12320   ;FROM CPM.
12322   ;
12400   ;
12500   KSTAT:  IN      STAPT   ;GET STATUS WORK
12600           CMA             ;INVERT IT FOR PROPER RETURN
12700           ANI     KDR     ;TEST KEYBOARD BIT
12800           RZ              ;ZERO IS NO CHARACTER RECEIVED
12900   ;
13000           JMP     CPMCMD  ;GO CHECK FOR MODE SELECT
13200   ;
13300   ;
13400   ;   THIS JUMP IS PART OF THE AUTO START UP CODE
13500   ;
13600           DB      0       ;VERIFY ADDR=C037
13700           JMP     INIT    ;THIS SHOULD BE C038
```

```
29250   ;
29260   ;       THIS LOOKS FOR END OF SCREEN BY COMPARING THE NEXT
29270   ;ADDRESS AFTER THE LAST SCREEN ADDRESS. THIS WORKS FINE IF
29280   ;SOLOS ISN'T AT 0F000H. THEN THE CODE SHOULD BE CHANGED
29290   ;TO THE FOLLOWING
29292   ;
29294   ;       CPI     0
29296   ;       JNZ     ERAS1
29298   ;
29299   ;THE CPI 0F0H IS FOR SOLOS AT 0E000H.
29300   ;
29310   ;
29400   PERSE:  LXI     H,VDMEM ;POINT TO SCREEN
29500           MVI     M,80H+' '       ;THIS IS THE CURSOR
29600   ;
29700           INX     H       ;BUMP   1ST
29800   ERAS1:                  ;LOOPS HERE TO ERASE SCREEN
29900           MVI     M,' '   ;BLANK IT OUT
30000           INX     H       ;NEXT
30100           MOV      A,H    ;SEE IF END OF SCREEN YET
30200           CPI     0F0H    ;COMPARE WITH FIRST ADD. AFTER ENC
30300           JC      ERAS1   ;NO--KEEP BLANKING
30400           STC             ;CARRY WILL SAY COMPLETE ERASE
30500   ;
```

```
38700   ;
38800   ;  ROUTINE TO PROCESS A CARRIAGE RETURN
38900   ;
39000   PCR:    XRA     A       ;REWIND TO BEGINNING OF LINE
39010           NOP
39020           NOP
39100   ;
39200   ;ORIGINAL HAD 'CALL CLINE' INSTEAD OF 'XRA A' AS SHOWN
39300   ;ABOVE. THIS CAUSED DISK COMMANDS TO DISAPPEAR FROM THE
39400   ;SCREEN AS CP/M MUST OUTPUT TWO 'CR' CHARACTERS.
39500   ;
39510   ;THE TWO NOP'S HAVE BEEN ADDED TO KEEP THE ADDRESSES OF THE
39520   ;FOLLOWING SUBROUTINES THE SAME AS IN THE ORIGINAL VERSION.
39530   ;
39600           JMP     PCUR    ;AND STORE THE NEW VALUE
39700   ;
```

```
47700   ;       ***** START UP SYSTEM *****
47800   ;
47900   ; CLEAR SCREEN AND THE FIRST 256 BYTES OF GLOBAL RAM
48000   ; THEN JUMP TO "INCPM". IT WILL THEN FINISH THE INITIALIZATION
48010   ;AND JUMP TO CPM COLDBOOT. MOVED TO MAINTAIN ADDRESSES.
48020   ;
48100   ;
48200   STRTA:  XRA     A
48300           MOV     C,A
48400           LXI     H,SYSRAM        ;CLEAR THR FIRST PAGE
48500   ;
48600   CLERA:  MOV     M,A
48700           INX     H
48800           INR     C
48900           JNZ     CLERA
49000   ;
49100           LXI     SP,SYSTP        ;SET UP THE STACK FOR CALL
49200           JMP     INCPM           ;GO INIT TO CPM
49300   COMN1:  XRA     A
49400           OUT STAPT       ;BE SURE TAPES ARE OFF
49500           STA     OPORT
49600           STA     IPORT
49700   ;
```

```
84800   ;       ***** TERMINAL COMMAND *****
84900   ;
85000   ;       THIS ROUTINE GETS CHARACTERS FROM THE SYSTEM KEYBOARD
85100   ; AND OUTPUTS THEM TO THE SELECTED OUTPUT PORT.  IT IS
85200   ; INTENDED TO CONFIGURE THE SOL AS A STANDARD VIDEO
85300   ; TERMINAL.  COMMAND KEYS ARE NOT OUTPUT TO THE OUTPUT
85400   ; PORT BUT ARE INTERPRETED AS DIRECT SOL COMMANDS.
85500   ; THE MODE COMMAND, RECEIVED BY THE KEYBOARD, PUTS THE SOL
85600   ; IN THE COMMAND MODE.
85700   ;
85710   ;       AND AFTER ALL THAT ,THIS SPACE IS USED FOR NIFFTY
85712   ;LITTLE PATCHES AND THE TERMINAL MODE ELIMINATED. NOTE
85714   ;THAT THE PROPER NUMBER OF NOP'S MUST BE INSERTED TO
85716   ;TO FILL THE ADDRESSES IF THE .PHASE ISN'T USED.
85800   ;
85900   ;
86000   TERM:   JMP     COMN1   ;GOTO CMD
86100   ;
86200   CPMCMD: IN      KDATA
86300           CPI     80H             ;CHECK FOR "MODE SELECT"
86400           JZ      COMN1           ;IF SO, GOTO CMD
86500           RET
86600   ;
86700   INCPM:  CALL    PERSE
86800           XRA     A
86900           OUT     STAPT
87000           STA     OPORT
87100           STA     IPORT
87200           JMP     0F800H          ;GOTO CPM
87300   ;
87400   ;
89600           .DEPHASE
89700           .PHASE  SOLOS+3BFH
89800   ;
```

```
91400   ;
91500   DUMP:   CALL    SCONV   ;SCAN TO FIRST ADDRESS AND CONVERT IT
91600           PUSH    H       ;SAVE THE VALUE
91700           CALL    PSCAN   ;SEE IF SECOND WAS GIVIN
91800           POP     D
91900           XCHG            ;HL HAS START, DE HAS END
```

```
41800   ;
41900   ;
42000   PCLOS:  CALL    LFCB    ;GET CONTROL BLOCK ADDRESS
42100           RZ              ;WASN'T OPEN, CARRY IS SET FROM LFCR
42200           ORA     A       ;CLEAR CARRY
42300           INR     A       ;SET CONDITION FLAGS
42400           MVI     M,0     ;CLOSE THE CONTROL BYTE
42500           RZ              ;WE WERE READING...NOTHING MORE TO DO
42600   ;
42700   ; THE FILE OPERATIONS WERE "WRITES"
```

CON'T FROM PAGE 8: SOLOS MODIFICATION FOR CP/M

```
42800   ;
42900   ; PUT THE CURRENT BLOCK ON THE TAPE
43000   ; (EVEN IF ONLY ONE BYTE)
43100   ; THEN WRITE AN END OF FILE TO THE TAPE
43200   ;
43210   ;
43212   ;          NOTE THE ORIGINAL MISTAKE THAT HAS BEEN PUT
43214   ;BACK IN. A NOP COULD ALSO BE USED BUT WHAT THE HECK.
43216   ;
43300   ;
43400           INX     H
43500           INX     H
43600           MOV     A,M     ;GET CURSOR POSITION
43610           MOV     A,M     ;TO BE ORIGINAL
43700           CALL    PLOAD   ;BC GET HEADER ADDRESS, DE BUFFER ADDRESS
43800           PUSH    B       ;HEADER TO STACK
43900           LXI     H,BLKOF ;OFFSET TO BLOCK SIZE
44000           DAD     B
44100           ORA     A       ;TEST COUNT
44200           JZ      EOFW    ;NO BYTES...JUST WRITE EOF
44300   :
```

CROSS REFERENCE FOR BASIC VARIABLES
BY Tom Cardoso

In your December newsletter you advised that a cross-reference program for ECBasic would be forthcoming, but I haven't seen it yet so I thought I would send you mine to publish if you wish. I wrote it a couple of years ago and although it's very functional, I'm not particularly proud of its speed. The program will check for all ECBasic variables and list them alphabetically with all their line number references. To give you an idea of time, the program will process itself in about 8 mins. and Smart-mouth Blackjack (a 16k program) in about 30 min., not great, but when its all you have it's OK. If anyone cares to offer suggestions on improving the speed, short of a complete re-write, the comments would be most welcome. The program is set to use OP=3 as a printer output pseudoport. If other ports are implemented, change line 860 appropriately.

I continue to relish the newletter and offer my services to you if there is some way you feel I can help.

Tom Cardoso
1469 Farington Dr.
Naperville, Ill. 60540
312-420-0166

```
 10 REM     XREF
 20 REM     VERSION 5.2 ... 7/81
 30 REM
 40 REM     CROSS REFERENCE PROGRAM FOR BASIC VARIABLES
 50 REM     WRITTEN FOR SOL EXTENDED CASSETTE BASIC
 60 REM     WRITTEN BY TOM CARDOSO ... 312-420-0166 ... 6/79
 70 REM
 80 REM     This program requires that the BASIC program
 90 REM     statements be saved on tape or disk in TEXT mode,
100 REM     using a filename other than 'XREF'.  This program
110 REM     will read that text file, extract the variables
120 REM     and list them alphabetically, together with
130 REM     all their line references.
140 REM
150 REM     This program works on EXTENDED CASSETTE BASIC
160 REM     programs only, with variables from 'A' TO 'Z9$'.
170 REM     It will not extract double-letter variables.
180 REM
190 REM     Program will print an asterisk '*' for each line
200 REM     read and processed so you can follow progress.
210 REM
220 REM     The DIM value of V$ below can be lowered to suit
230 REM     smaller memories.
240 REM
250 DIM V$(20000),A$(100): SET OP=0
260 PRINT "&K": CURSOR 3,0: PRINT TAB(15);"VARIABLE CROSS REFERENCE TABULATOR"
270 CURSOR 6,0: INPUT (1,0)"Do you want this to go to PRINTER?  Y or N ... ",Z9$
280 IF Z9$="y" THEN  LET Z9$="Y"
290 IF Z9$<>"Y" THEN  LET Z9$="N"
300 PRINT : INPUT (7,0)"Enter Filename to READ from ... ",N1$
310 LET N1$=N1$+",T"
320 PRINT : PRINT "Reading file and processing ... patience please ... ": PRINT
330 FILE #1;N1$,1
340 READ #1;A$: CLOSE #1: GOTO 860
350 PRINT "*";
360 LET X=LEN(A$): LET N0$=A$(1,5): LET A$=A$(7,X): LET X=LEN(A$)
370 IF A$(1,3)="REM" THEN 340
380 IF A$(1,4)="DATA" THEN 340
390 FOR Y=1 TO X
400   LET B$=A$(Y,Y)
410   IF ASC(B$)<>34 THEN 450
420   FOR Y0=Y+1 TO X
430     LET B0$=A$(Y0,Y0): IF ASC(B0$)=34 THEN  LET Y=Y0: LET Y0=X
440   NEXT
450   IF Y<X THEN  LET C$=A$(Y+1,Y+1) ELSE  LET C$=""
460   IF Y<X-1 THEN  LET M$=A$(Y+2,Y+2): LET G$=M$: IF M$<>"$" THEN  LET M$=""
470   IF Y>1 THEN  LET D$=A$(Y-1,Y-1)
480   IF B$>="A" THEN  IF B$<="Z" THEN 500
490   GOTO 830
500   IF D$="&" THEN 830
510   IF C$="" THEN  IF D$="," THEN 810
520   IF B$+C$+G$="REM" THEN  LET Y=X: GOTO 830
530   IF C$="" THEN  IF D$=" " THEN 820
540   IF C$="" THEN 660
550   IF C$<>"=" THEN 620
560   IF D$+B$="OP" THEN 830
570   IF D$+B$="LL" THEN 830
580   IF D$+B$="DB" THEN 830
590   IF D$+B$="DS" THEN 830
600   IF D$+B$="IP" THEN 830
610   IF D$+B$="ML" THEN 830
620   IF C$>="0" THEN  IF C$<="9" THEN 630 ELSE 650 ELSE 650
630   IF D$>="0" THEN  IF D$<="9" THEN 830
640   GOTO 810
650   IF C$<>" " THEN 740
660   IF D$=" " THEN 820
670   IF D$="<" THEN 820
680   IF D$=">" THEN 820
690   IF D$="=" THEN 820
700   IF D$="+" THEN 820
710   IF D$="-" THEN 820
720   IF D$="/" THEN 820
730   IF D$="*" THEN 820 ELSE 830
740   IF C$="(" THEN  IF D$<"A" THEN 820 ELSE 830
750   IF C$="$" THEN 810
760   IF C$>="A" THEN  IF C$<="Z" THEN 830
770   IF C$>="a" THEN  IF C$<="z" THEN 830
780   IF C$<"0" THEN 820
790   IF C$>"9" THEN 820
800   GOTO 830
810   LET V$=V$+B$+C$+M$+" "+N0$: GOTO 830
820   LET V$=V$+B$+M$+" "+N0$: GOTO 830
830 LET M$="": NEXT Y
```

CON'T FROM PAGE 9: - CROSS REF FOR BASIC VARIABLES

```
840 GOTO 340
850 REM     OUTPUT ROUTINES FOLLOW
860 IF Z9$="Y" THEN  SET OP=3: SET LL=128: PRINT CHR(29)
870 PRINT : PRINT TAB(18);"VARIABLE CROSS REFERENCE TABLE FOR `";N1$;"'"
880 PRINT : PRINT "VAR.";TAB(10);"LINE NUMBER REFERENCES": PRINT
890 PRINT : PRINT
900 LET S=65,A1$="",Y=0
910 LET D$=CHR(S)+A1$+A2$+" ": IF D$="Z0 " THEN  LET Y=1
920 IF D$="Z0$ " THEN  LET Y=1
930 SEARCH D$,V$,X1
940 IF D$="Z9$ " THEN  IF X1=0 THEN  PRINT CHR(30): SET OP=0: PRINT : END
950 IF X1=0 THEN  LET S=S+1 ELSE 1010
960 IF S>90 THEN  LET S=65 ELSE 910
970 IF Y=0 THEN  LET A1=0 ELSE  LET A1=A1+1
980 IF A2$="$" THEN  IF A1>9 THEN  PRINT CHR(30): SET OP=0: PRINT : END
990 IF A1>9 THEN  LET A2$="$": GOTO 900
1000 LET A1$=STR(A1): GOTO 910
1010 ON LEN(D$)-1 GOTO 1020,1030,1040
1020 LET E$=V$(X1+2,X1+6)+", ": GOTO 1050
1030 LET E$=V$(X1+3,X1+7)+", ": GOTO 1050
1040 LET E$=V$(X1+4,X1+8)+", "
1050 IF D$=F$ THEN 1100 ELSE  LET Y0=1
1060 ON LEN(D$)-1 GOTO 1070,1080,1090
1070 PRINT : PRINT : PRINT D$;"  : ";: GOTO 1110
1080 PRINT : PRINT : PRINT D$;" : ";: GOTO 1110
1090 PRINT : PRINT : PRINT D$;": ";
1100 IF Z9$="N" THEN  IF Y0=6 THEN  PRINT : PRINT "      ";: LET Y0=1: GOTO 1120
1110 IF Y0=10 THEN  PRINT : PRINT "      ";: LET Y0=1
1120 PRINT E$;: LET Y0=Y0+1
1130 LET F$=D$
1140 ON LEN(D$)-1 GOTO 1150,1190,1230
1150 IF X1=1 THEN  IF LEN(V$)=7 THEN 1270
1160 IF X1=LEN(V$)-6 THEN  LET V$=V$(1,X1-1): GOTO 930
1170 IF X1=1 THEN  LET V$=V$(X1+7,LEN(V$)): GOTO 930
1180 LET V$=V$(1,X1-1)+V$(X1+7,LEN(V$)): GOTO 930
1190 IF X1=1 THEN  IF LEN(V$)=8 THEN 1270
1200 IF X1=LEN(V$)-7 THEN  LET V$=V$(1,X1-1): GOTO 930
1210 IF X1=1 THEN  LET V$=V$(X1+8,LEN(V$)): GOTO 930
1220 LET V$=V$(1,X1-1)+V$(X1+8,LEN(V$)): GOTO 930
1230 IF X1=1 THEN  IF LEN(V$)=9 THEN 1270
1240 IF X1=LEN(V$)-8 THEN  LET V$=V$(1,X1-1): GOTO 930
1250 IF X1=1 THEN  LET V$=V$(X1+9,LEN(V$)): GOTO 930
1260 LET V$=V$(1,X1-1)+V$(X1+9,LEN(V$)): GOTO 930
1270 PRINT CHR(30): SET OP=0: PRINT : END
1280 REM     END PROGRAM
```

STATIC ELECTRICITY CONTROL

From time to time, when I am wearing certain shoes or clothes, I zap my Sol with static electricity. The program crashes (goes crazy in some way, such as writing garbage on the screen or on the disk) and/or the keyboard does funny things (like turning all of the 'red LED indicators on and sending an extra character to the screen). Sometimes this does no permanent damage to the data I'm working on, and other times it just destroys it and I have to reconstruct the data from a backup disk. If you suffer from the same scary problem, you may want to try some of these solutions.

The idea is to dissipate the electric charge before it builds to the point of jumping to the ground through whatever you touch. You should avoid wearing those clothes that tend to hold static charges, such as synthetics, plastic soles, wool, etc. Plain cotton is the best. But sometimes this isn't practical, as in an office situation.

The most practical approach is to put a grounding surface under your chair or under your feet. That is, a fairly good conductor connected to ground should be in contact with you. I say "fairly good" because it is not safe to be directly connected to ground with an excellent conductor (such as a wire), just in case you come into contact with a power source through a short circuit in your equipment.

Static control mats that meet these requirements are made by 3M company under the brand name Velostat. They come in various sizes, for stand-up work stations, for under-desk and under-chair situations, and for larger area (4' x 8') protection. They are made of a plastic that contains enough of a conductive material to carry off the charges through a grounding wire before they can become harmful to electronics. The grounding cord has a 1 megohm resistor in it to protect against dangerous electrocution in the event of electrical fault. The mats are reversible, so that if they tend to become warped through repeated use, the mat can be flipped over.

The mats are not cheap, but the loss of valuable data or destruction of static sensitive IC's could be more expensive. A desk-size, comparable to the plastic mats commonly used to protect carpets under secretarial chairs, costs about $160. They are available from business machine stores and computer supply dealers.

Another static control item is a spray that can be applied to carpets. This works, but it wears off and must be resprayed periodically. The solvent in the spray is some sort of hydrocarbon, and I'm sure repeated inhalation of this is not good for your body. It isn't cheap either, in the long run, because it must be done again and again. It's best use is to provide relief from the annoying ZAP you get when shuffling across a carpet and touching a metal object. The material I have used is called LAB NO-SHOCK made by Lab Automated Chemicals, a division of American Chemmate Corporation, Howard & West Streets, Baltimore, Maryland 21230, Phone (301) 752-2610. Building maintenance suppliers have this material.

A third solution, which I haven't tried, would be to attach a wire to a metal bracelet, and put a 1 megohm resister in the circuit to a nearby grounding point (not the computer of course). This would be annoying, but it should do the job.

Velostat Electrically Conductive Floor Mats drain static charge from personnel harmlessly to ground. Static charge is removed as personnel step on Velostat Mats during the performance of their everyday office functions. The mat prevents static caused loss of data and alteration of memory in computers and electronic office equipment. Because Velostat Mats are grounded, they can never be over-loaded with static charge and do not rely on room humidity to work.

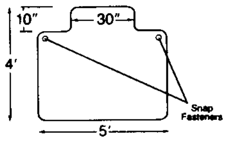
Velostat 1850 Series Floor Mats are made of a tough, durable, high impact conductive material. The material is hard and rigid so castered chairs easily roll over it — even if the mat is laid on plush, pile carpeting. The conductivity of Velostat Floor Mats is an inherent material property unaffected by age and not dependent on room humidity to provide protection. All mats come complete with ground cords.

Simply place the mat where the operator must step on it to reach the keyboard. The mat will provide protection in such key areas as these:

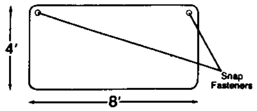
- Minicomputers
- Microcomputers
- Word Processing Equipment
- Data Entry Equipment
- Computer Printers
- Disc and Tape Drives

Dimensions:

Type: 1853
(⅛" thickness)



Type: 1854
(⅛" thickness)



PROGRAM TO RUN PTC CASSETTE PROGRAMS UNDER CP/M

BY Jim Dailey and Deslar K. Patten

```
PTOCPM.DOC   TRANSFERS PROC. TECH OBJECT CODE
             FILES TO CPM FILES WITH RELOCATOR.

PTOCPM.ASM   M80 (MICROSOFT) ASSEMBLY FILE

PTOCPM.COM   OBJECT PROGRAM FOR CPM

PTOCPM.PRN   LISTING FILE WITH XREF

FUNCTION:
        TO CONVERT MACHINE CODE FILES FROM
PROC TECH TAPES TO CPM FILES WITH A RELOCATOR
PACKAGE ATTACHED SO THEY CAN BE RUN UNDER CPM
AT THEIR ORIGINAL LOCATION.

HARDWARE DEPENDENCIES:
        PROC TECH SOL COMPUTER WITH SOLOS MODULE
AND MICROMATION DOUBLER. 2" DISK DRIVE AND TAPE
RECORDER ALSO NEEDED.

SOFTWARE DEPENDENCIES:
        THIS WORKS WITH SOLOS AT ITS ORIGINAL
LOCATION.THE ASSEMBLY LISTING HAS BEEN CHANGED SO
IT CAN BE USED WITH SOLOS AT A DIFFERENT ADDRESS
BY CHANGING "SOLOS EQU" AND REASSEMBLING.

        IF YOUR CPM COLD BOOT ADDRESS ISN'T AT F800,
THEN CHANGE THE "CPMCBT EQU" TO THE ADDRESS USED
IN YOUR SYSTEM AND REASSEMBLE.


        THIS PROGRAM WAS WRITTEN BY JIM BAILEY,
AND MODIFIED BY DES PATTEN ON 2-22-81.THE SOURCE
LISTING  CALLED PGMCPYC9.ASM WAS IN VOL.3 # 2 OF
THE PROTEUS NEWS.

        STEP BY STEP PROCEEDURE

1.BOOT UP IN CPM.

2.TYPE "PTOCPM" AND CARRIAGE RETURN.
        THIS WILL RELOCATE THE PROGRAM AT C900H,
CALL THE INITIALIZATION ROUTINE AT C900H, AND
THEN JMP TO THE SOLOS COMMAND MODE.

3.AFTER GETTING THE COMMAND PROMPT, TYPE IN ;

        GET  <FILENAME> AND A CARRIAGE RETURN

AS YOU WOULD NORMALLY DO TO GET A TAPE FILE
INTO MEMORY.

4.WHEN THE TAPE STOPS THE CONSOLE WILL PRINT
OUT THE FOLLOWING;

        <FILENAME> <TYPE> <XXXX> <YYYY>

WHERE XXXX= STARTING ADDRESS
      YYYY= LENGTH OF FILE

5.TYPE IN THE FOLLOWING;

        CD  AND A CARRIAGE RETURN


6.THE NUMBER OF PAGES REQUIRED FOR 'SAVE'
BE PRINTED OUT, AND THEN THE CPM PROMPT
WILL APPEAR.

7.WHEN THE CPM PROMPT APPEARS, TYPE IN THE
FOLLOWING;

        SAVE  <PAGES>  <FILENAME.TYPE>


8. DONE!

        WHEN THE SAVED PROGRAM IS CALLED BY CPM,
IT WILL BE LOADED INTO THE 100H FILE AREA.THEN
THE JUMP INSTRUCTION AT THE BEGINNING OF THE FILE
WILL EXECUTE THE RELOCATOR PACKAGE AT THE END OF
FILE. THIS WILL MOVE THE PROGRAM TO IT'S  ORIGINAL
LOCATION AND RUNS IT.

GOOD LUCK!
```

```
;PTOCPM.ASM      SOURCE FOR PTOCPM.COM, A PROGRAM
;                TO LOAD PROC TECH OBJECT CODE TAPES
;                ON CPM DISKS.
;
;WRITTEN BY JIM BAILEY
;PUBLISHED IN PROTEUS  VOL. 3  #2
;MODIFIED BY DES PATTEN  2-22-81, 3-9-81
;
;SOFTWARE DEPENDENCIES:   M80 (MICROSOFT) ASSEMBLER
;
;MODIFICATION PURPOSE:    TO HAVE A RESIDENT CPM FILE
;                         OF PGMCPY9.ASM.
;
;SOLOS LOCATION IS ASSUMED AT C000H. HOWEVER SINCE
;ALL "SOLOS EQUATES" ARE REFERANCED TO "SOLOS", ALL
;THAT IS REQUIRED IS TO CHANGE "SOLOS EQU" FOR OTHER
;LOCATIONS OF SOLOS.
;
;
;
;
;       EQUATES          TABLE
;
CPMCBT  EQU     0F800H  ;CPM COLD BOOT LOCATION, THIS CAN
;                        BE CHANGED FOR YOUR SYSTEM.
;
SOLOS   EQU     0C000H  ;CHANGE THIS IF SOLOS IS MOVED
;                        IN YOUR SYSTEM
;
SOUT    EQU     SOLOS+19H       ;SOLOS OUTPUT
SINP    EQU     SOLOS+1FH       ;SOLOS INPUT
AOUT    EQU     SOLOS+1CH       ;SOLOS ASSIGNED OUTPUT
CUTAB   EQU     SOLOS+83CH      ;1ST SPOT IN CUS. COM. TBL
CRLF    EQU     SOLOS+2F9H      ;OUTPUT CR &LF
ADOUT   EQU     SOLOS+3E8H      ;OUTPUT HL AS HEX NUMBER
COMN1   EQU     SOLOS+1C0H      ;COMMAND MODE
PGMCPY  EQU     SOLOS+900H      ;PGMCPY LOCATION
BLOCK   EQU     SOLOS+823H      ;FILE LENGTH
LOADR   EQU     SOLOS+825H      ;FILE START
;
START   EQU     0000H
FILER   EQU     103H    ;CPM FILE START+3
;
;
                ASEG
                ORG     100H
                JMP     RELOC   ;JUMP TO RELOCATOR USED FOR
                                ;PTOCPM
;
;
```

CON'T FROM PAGE 11: - PTC CASSETTE UNDER CP/M

```
;***********************************************
;
;THIS IS THE PROGRAM TO BE LOADED INTO C900 THAT
;DOES ALL THE WORK.IT IS MOVED BY THE RELOCATOR
;PACKAGE AT THE END OF THE PROGRAM.IT IS THE PGMCPY
;PROGRAM WRITTEN BY JIM BAILEY.
;
;THE MODIFICATION TO THIS PROGRAM HAS BEEN TO ADD
;THE JMP TO CPM COLD BOOT INSTEAD OF RETURN TO
;SOLOS COMMAND. 2-22-81
;
;THE ADDRESSES ARE NOW OBTAINED FROM THE FILE HEADER
;STORED IN RAM.ALSO THE NUMBER OF PAGES TO BE SAVED
;ARE OUTPUT IN DECIMAL SO NO CONVERSION IS REQUIRED.
;3-9-81
;
;***********************************************
;
                .PHASE  0C900H
;
; THIS PUTS CUSTOM COMMAND IN TABLE
;
                LXI     H,'DC'
                SHLD    CUTAB
                LXI     H,DISKIT
                SHLD    CUTAB+2
                RET
;
;THIS OBTAINS THE START ADD. AND FILE LENGTH FROM THE
;RAM FILE HEADER AND CALCULATES THE OLD AND NEW END.
;
DISKIT:         LHLD    LOADR           ;GET FILE START
                PUSH    H
                LHLD    BLOCK           ;GET FILE LENGTH
                XCHG                    ;PUT LENGTH IN DE
                POP     H               ;GET FILE START BACK
                PUSH    H               ;SAVE OLD START FOR LATER
                DAD     D               ;OLD BEGIN + LENGTH = OLD END
                PUSH    H               ;SAVE OLD END FOR NOW
                LXI     H,FILER         ;GET NEW BEGINNING
                DAD     D               ;NEW BEGIN+LENGTH=NEW END
                POP     D               ;GET OLD END BACK INTO DE
                PUSH    H               ;SAVE NEW END FOR LATER
;
;NOW H,L HAVE NEW END AND D,E HAVE OLD END
;STACK NOW HAS NEW END ON TOP AND OLD START JUST BELOW.
;THESE ARE USED BY 'RESET' AFTER 'MOVUM'.
;
;THIS MOVES THE PROGRAM TO 100H
;
MOVUM:          LDAX    D
                MOV     M,A
                DCX     D
                DCX     H
                MOV     A,C
                CMP     E
                JNZ     MOVUM
                MOV     A,B
                CMP     D
                JNZ     MOVUM
                LDAX    D
                MOV     M,A
;
;THIS PUTS THE CORRECT JUMP ADDRESSES INTO THE RELOCATOR
;PACKAGE
;
RESET:          POP     H
                INX     H
                SHLD    101H
                MVI     A,0C3H
                STA     100H
                SHLD    TRIPPR+1
                LXI     D,0AH
                DAD     D
                SHLD    FIRST+1
                SHLD    SECOND+1
                POP     H
                SHLD    TRIPPR+4
                LHLD    101H
;
;THIS INSTALLS LOCATOR PACKAGE AND OUTPUTS # OF PAGES.
;
                LXI     D,TRIPPR
                MVI     C,1DH
MORE:           LDAX    D
                MOV     M,A
                INX     H
                INX     D
                DCR     C
                JNZ     MORE
                CALL    CRLF
                MOV     A,H
                CALL    OUTDEC
                CALL    CRLF
                JMP     CPMCBT  ;GO TO CPM COLD START.
;
;THIS IS THE RELOCATOR PACKAGE
;THAT IS ATTACHED BY THE PROGRAM
;MOVED TO C900.
;
TRIPPR:         LXI     B,START
                LXI     D,START
                LXI     H,FILER
                PUSH    D
MOVIT:          MOV     A,M
                STAX    D
                INX     D
                INX     H
                MOV     A,L
                CMP     C
FIRST:          JNZ     MOVIT
                MOV     A,H
                CMP     B
SECOND:         JNZ     MOVIT
        LXI     H,SOLOS
                RET
;
;THIS ROUTINE OUTPUTS BINARY OR HEX AS A DECIMAL.
;ENTER WITH DATA IN A REG.
;USES A,B,C,D, AND E REGS.
;
OUTDEC:         LXI     D,0000H
OTDEC0:         SUI     64H
                JC      OTDEC1
                INR     E
                JMP     OTDEC0
OTDEC1:         ADI     64H
                PUSH    PSW
                MOV     A,E
                MVI     E,0
                ORA     A
                JNZ     OTDEC2
                DCR     D
                MVI     A,0F0H
OTDEC2:         ADI     30H
                MOV     B,A
                CALL    SOUT
                INR     D
                POP     PSW
```

CON'T FROM PAGE 12: - PTC CASSETTE UNDER CP/M

```
18600   OTDEC3:         SUI     0AH
18700                   JC      OTDEC4
18800                   INR     E
18900                   JP      OTDEC3
19000   OTDEC4:         ADI     0AH
19100                   PUSH    PSW
19200                   MOV     A,E
19300                   ORA     D
19400                   JNZ     OTDEC5
19500                   MVI     A,0F0H
19600                   JP      OTDEC6
19700   OTDEC5:         MOV     A,E
19800   OTDEC6:         ADI     30H
19900                   MOV     B,A
20000                   CALL    SOUT
20100                   POP     PSW
20200                   ADI     30H
20300                   MOV     B,A
20400                   CALL    SOUT
20500                   RET
20600   ;
20700   BUFPTR:         DS      2
20800   ;
20900                   .DEPHASE
21000   ;
21100   ;**********************************************
21200   ;
21300   ;THIS IS THE RELOCATOR PACKAGE USED BY PTOCPM
21400   ;TO MOVE THE ABOVE PROGRAM TO C900.WHEN PTOCPM
21500   ;IS CALLED BY CPM THE FOLLOWING TAKES PLACE;
21600   ;
21700   ;       1.PTOCPM IS FIRST LOADED INTO CPM AT
21800   ;       100H.
21900   ;       2.THE JMP AT 100H GOES TO THE RELOCATOR
22000   ;       PROGRAM FOLLOWING, WHICH MOVES THE C900
22100   ;       PROGRAM ABOVE, TO C900.
22200   ;       3.THEN IT CALLS THE FIRST ROUTINE AT C900
22300   ;       TO SET UP THE CUSTOM COMMAND TABLE.
22400   ;       4.IT THEN JUMPS TO THE COMMAND MODE OF
22500   ;       SOLOS.
22600   ;
22700   RELOC:          LXI     B,RELOC ;BEGINNING OF RELOCATOR
22800                   LXI     D,PGMCPY;THIS IS THE ADD. WHERE
22900   ;                               ;IT MOVES TO.
23000           LXI     H,FILER         ;THIS IS WHERE IT IS NOW.
23100                   NOP
23200   MLOOP:          MOV     A,M
23300                   STAX    D
23400                   INX     D
23500                   INX     H
23600                   MOV     A,L
23700                   CMP     C
23800                   JNZ     MLOOP
23900                   MOV     A,H
24000                   CMP     B
24100                   JNZ     MLOOP
24200                   CALL    0C900H  ;SETUP CUS COM TBL
24300                   JMP     COMN1   ;GO TO COMMAND MODE
24400                   NOP
24500   ;
24600   ;       THE DIFFERANCE BETWEEN THIS RELOCATOR AND
24700   ;THE RELOCATOR NORMALLY ADDED TO A CONVERTED
24800   ;PROGRAM, IS THE FOLLOWING;
24900   ;
25000   ;       1.THE PUSH D WAS CHANGED TO A NOP.
25100   ;       2.THE LXI H,C000 WAS REPLACED WITH A
25200   ;       CALL  C900.
25300   ;       3.THE RET WAS REPLACED BY A JMP  COMN1.
25400   ;
25500   ;       THIS PREVENTS THE PROGRAM FROM STARTING
25600   ;AT C900 AND THEN RETURNING TO CPM AT C90C, WHICH
25700   ;CAUSES THE MOST SPECTACULAR FIREWORKS I HAVE EVER
25800   ;SEEN.
25900   ;
26000   ;GOOD LUCK!
26100                   END
```

RANDOMIZE FOR PT BASIC
BY Richard Bjorndal

I WAS UNAWARE OF ANY NEED TO IMPLEMENT A RANDOMIZE FUNCTION FOR PT. BASICS. I AM UNABLE TO FIND ANY REPETITION IN THE RND FUNCTION PROVIDING ONLY THE RND(O) IS USED. ONCE 'RND' IS GIVEN A NUMERICAL ARGUMENT OTHER THAN 'O', ANYWHERE IN THE PROGRAN, A SEED CHAIN IS ESTABLISHED AND EVEN RND(O) BECOMES PREDICTABLE THEREAFTER. RICHARD'S MINI-PROGRAM WILL NOT FIX IT. RE-RUNNING THE PROGRAM WILL NOT FIX IT. SCRATCHING AND RE-LOADING THE PROGRAM WILL NOT FIX IT. REMOVING ANY STATEMENT IN THE PROGRAM THAT GIVES A NUMERICAL ARGUMENT OTHER THAN 'O' TO RND WILL FIX IT.

EXAMPLE: TRY THIS.....

```
10 FOR l=1 TO 6:PRINT RND(O);:NEXT I
   (RUN THAT PROGRAM I SEVERAL TIMES, OBSERVING RANDOMNESS)
              ....NOW ADD THIS LINE...
20 PRINT RND(4)
   (RUN AGAIN I, SEVERAL TIMES ..)
```

ON THE SECOND RUN, AFTER RND(4) HAS BEEN 'SEEN', THE RND(O) FUNCTION BECOMES REPETITIVE. REMOVE 20 AND RUN AGAIN AND ALL IS RANDOM. .....OR AM I MISSING A POINT?

HOWEVER, APPLESOFT AND THE BASICS ON 'SOFTCARD' CAN MAKE GOOD USE OF RICHARD'S PROGRAM. THEY REQUIRE SEEDING.

MURRAY MACKENZIE,
38 INNISWOOD DR.
SCARBOROUGH,M ONT.,
CANADA. M1R 1E5.

EDITOR'S NOTE:
The problem with use of RND(Ø) is that the next time you load BASIC and run your program it will behave exactly the same as it did the last session, since the number generator will start again at the beginning.
Seeding allows you to get a different series of random numbers in each unrelated session. But not all numbers are good as seeds. Depending upon the pseudo-random number generator algorithm, certain seeds will produce a relatively short series of unique numbers before beginning to repeat. Perhaps someone who has the BASIC source code (Proteus item P12 or P13) and who knows more about random number generators could make some recommendations for getting good seeds.

PATCHES FOR PT DISK BASIC (HIGH VERSION)

BY Ben C. Stapleton, Jr.

Some of the patches I've made to PT Disk Basic (High Version) might be of interest to other members.

This version gives BASIC files a protection attribute when saved to disk which makes them difficult to list with the FILES command of PTDOS.

With some clues on where and what to look for, provided by Kris Fincher of Computer Port, Arlington, Texas (Nice People), a patch to remove that feature has been found.

My Sol/Helios is used for business as well as pleasure and one small irritant for me has been the inability to defeat the Mode Select key. No matter how carefully programs are error-trapped, the operator can exit Basic by accidentally hitting the Mode Select key. A patch to change Mode Select to Control C (or any other code you choose) is shown.

The bug with the REM statement mentioned by Bob Werner in the Jan/Feb 1980 issue of Proteus is also present in DBASIC. His fix works quite well with only the addresses being changed. One small hitch occurs when using the CAT command. The normal command to CATalog programs from Disk Unit 1 is: CAT /1. After this patch is in place, a syntax error will result unless the space is omitted. ie CAT/1.

I select the Internal Screen Driver when initializing DBASIC. To output to my printer, the statement: SET OF="PRNTR" is used, and SET OF=#0 returns output to the screen. It also clears the screen, which is not always desirable. This can be defeated.

Lastly, there are 510 bytes of memory unused by BASIC after FDEO. I put the Screen Copy routine from Dr. Dobbs Journal in this space and now can output an exact copy of the screen to my printer at any time from within a BASIC program.

<<< PATCHES FOR PT DISK BASIC (HIGH VERSION) >>>

Allow creation of BASIC files without Attributes:

```
     Change ED8E from 38 to 00    (affects C type files, 05)
     Change ED81 from 08 to 00    (affects T type files, 06)
```

Change from Mode Select to Control C:

```
     Change 057C from 00 to 03
     ENter @ 05A5: C3 E0 FD
     ENter @ FDE0: FE 03 CA 80 05 C3 A8 05
     ENter @ 063D: C3 F0 FD
     ENter @ FDF0: FE 03 CA 80 05 C3 40 06
```

Allow blanks after REM statement:

```
     ENter @ D161: 13 1A B7 FA C1 D1 47 CD 56 D3 23 B8
                   CA 61 D1 00
```

To prevent SET OF=#0 from clearing the screen:

```
     Use:     POKE 2077,148
              SET OF=#0
              POKE 2077,138
```

The listing for the Screen Copy routine is attached.

Now, I've got a question for some of the real software Pros out there. Has anyone thought about writing a device driver that would allow PTDOS to access a Hard Disk as though it was simply a super-large floppy? I'm interested in storage capacity, not speed. Also, the drive should be a cartridge type so it can be backed-up. I'd appreciate any thoughts at all on this.

Thanks for listening,

Ben

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

```
                         *     SCREEN IMAGE HARD COPY FOR THE SOL-20
                         *
                         *     FROM THE APRIL 1977 ISSUE OF:
                         *     DR. DOBB'S JOURNAL OF COMPUTER CALISTHENICS & ORTHODONTI
                         *     BOX E, MENLO PARK, CALIFORNIA 94025
                         *
                         *     ORIGINALLY WRITTEN FOR THE VDM-1 BY DAN S. PARKER
                         *
                         *     PATCHED FOR THE SOL-20 BY BEN STAPLETON, JR.
                         *
                         *     ORIGIN OF OFE10H WAS CHOSEN TO ALLOW THIS ROUTINE TO FIT
                         *     INTO UNUSED SPACE OF THE `HIGH' VERSION OF PT DBASIC.
                         *     LOAD "SCRNCOPY" INTO YOUR INITIALIZED BASIC BEFORE YOU
                         *     IMAGE IT.
                         *
                         *     "SCRNCOPY" CAN BE CALLED FROM BASIC ANYTIME YOU NEED A
                         *     PRINTOUT, BY THE STATEMENT: X=CALL(65040)
                         *
                         *                  "SCRNCOPY"
                         *
    FE10                           XEQ      START
 FE10                    START     ORG      OFE10H
 FE10 F5                           PUSH     PSW          ;SAVE REGISTERS
 FE11 E5                           PUSH     H
 FE12 D5                           PUSH     D
 FE13 C5                           PUSH     B
 FE14 3E 01                        MVI      A,01H
 FE16 32 07 C8                     STA      OPORT        ;SET OUTPUT TO SERIAL PORT
 FE19 21 00 CC                     LXI      H,OCC00H     ;SCREEN MEMORY ADDRESS
 FE1C 3A C8 08                     LDA      BOTL         ;TEXT LINE OFFSET
 FE1F FE 00                        CPI      00H
 FE21 CA 2F FE                     JZ       CONT
 FE24 11 40 00                     LXI      D,0040H      ;PUT CHARS/LINE IN D&E
 FE27 19                 CONT1     DAD      D            ;ADD TO H&L
 FE28 3D                           DCR      A
 FE29 CA 2F FE                     JZ       CONT         ;START OF TEXT IN H&L?
 FE2C C3 27 FE                     JMP      CONT1        ;CONTINUE IF NOT
 FE2F 11 11 40           CONT      LXI      D,4011H      ;CHARS & LINES IN D&E
 FE32 CD F9 C2           NXTLN     CALL     CRLF         ;GET CARRIAGE RET & LINE FEED
 FE35 7C                           MOV      A,H          ;
 FE36 FE D0                        CPI      0D0H         ;STILL WITHIN SCREEN MEMORY?
 FE38 CC 4D FE                     CZ       RESET        ;RESET H&L IF NOT
 FE3B 1D                           DCR      E            ;DECREMENT LINE COUNT
 FE3C CA 51 FE                     JZ       CLOSE        ;DONE
 FE3F 16 40                        MVI      D,040H       ;RESET CHARACTER COUNT
 FE41 46                 CONT2     MOV      B,M          ;GET CHARACTER
 FE42 CD 19 C0                     CALL     WRITE        ;AND OUTPUT IT
 FE45 23                           INX      H            ;INCREMENT MEMORY POINTER
 FE46 15                           DCR      D            ;DECREMENT CHARACTER COUNT
 FE47 CA 32 FE                     JZ       NXTLN        ;START NEW LINE IF DONE
 FE4A C3 41 FE                     JMP      CONT2        ;OTHERWISE NEXT CHARACTER
 FE4D 21 00 CC           RESET     LXI      H,OCC00H     ;RESTORE H&L TO START OF SCREEN
 FE50 C9                           RET      .            ;MEMORY
                         *
 FE51 3E 00              CLOSE     MVI      A,00H
 FE53 32 07 C8                     STA      OPORT
 FE56 C1                           POP      B            ;RESTORE REGISTERS
 FE57 D1                           POP      D
 FE58 E1                           POP      H
 FE59 F1                           POP      PSW
 FE5A C9                           RET      .            ;RETURN TO CALLING PROGRAM
                         *
                         *                  I/O  UTILITIES
                         *
    C2F9                 CRLF      EQU      0C2F9H
    C807                 OPORT     EQU      0C807H       ;"01" IS PRINTER
    C019                 WRITE     EQU      0C019H       ;"SOUT" IN SOLOS
    08C8                 BOTL      EQU      08C8H        ;LINE # BEGINNING OF TEXT
                         * BOTL EQU 0C80AH FOR SOLOS VIDEO DRIVER
                         *          08C8H  FOR BASIC INTERNAL DRIVER
```

LOAN CALCULATION PROGRAM
BY David Armstrong

As promised, here is another program from the Armstrong Corporation SOL*STAR library. SOL owners with North Star disk drives can run it directly; PT BASIC users will have to modify it slightly with respect to the print formatting statements and the INCHAR$ function, etc.

The program has proven very useful for loan and mortgage calculations. Using it, I was able to catch an error the bank made on my last car loan! The program is easy to use, and is self-documenting. The user is prompted to supply five loan parameters -- principal, interest rate, number of payments per year, term in years, and regular payment. Any of these values can be calculated by simply leaving out that value (making a null entry) when prompted for it, and filling in the others.

Coming up for the next issue of PROTEUS is a Print Spooler I/O package for the North Star DOS, and a simple assembly language routine to send the SOL's video display contents to the printer each time the LOAD key is pressed. If there are any special program application areas that PROTEUS readers would like to see for the SOL/North Star system, let me know.

Sincerely,

Wm. David Armstrong

Wm. David Armstrong

P.S. The following function can be substituted for the INCHAR$ function specified in the program listing supplied:

```
600 DEF FNB$(Z)
610 IF Z0 THEN 630 \ Z0=51456
620 FOR I=Z0 TO Z0+6 \ READ Z \ FILL I,Z \ NEXT
630 Z=CALL(Z0) \ Z=INP(252)
640 RETURN CHR$(Z) \ FNEND
650 DATA 205,31,192,202,0,201,201
```

```
**************************************************************
      L O A N   C A L C U L A T I O N   P R O G R A M
**************************************************************


5  REMEMBER TO CREDIT WM. DAVID ARMSTRONG - (C)1981
10 PRINT CHR$(11), \ FOR I=1 TO 63 \ PRINT "*", \ NEXT \ PRINT
20 PRINT TAB(8),"L O A N   C A L C U L A T I O N   P R O G R A M"
30 PRINT TAB(15),"By Wm David Armstrong - (C)1981"
40 FOR I=1 TO 63 \ PRINT "*", \ NEXT \ PRINT  \ PRINT  \ DIM A$(100),A(5) \ A$="0"
50 PRINT "INSTRUCTIONS: Enter data as prompted. A null entry ",
60 PRINT "(RETURN key" \ PRINT "only) will cause that value to be ",
70 PRINT "computed." \ GOSUB 250
80 FOR A=1 TO 5 \ GOSUB 290 \ A(A)=X
90 IF X<0 THEN N=N+1 \ PRINT FNA$(18,A+8), \ NEXT
100 IF N>1 THEN 230 \ FOR X=1 TO 5 \ IF A(X)<0 THEN EXIT 130 \ NEXT
110 PRINT FNA$(0,13) \ PRINT "*** NO CALCULATION REQUESTED ***",FNA$(0,16),
120 FOR I=1 TO 2000 \ NEXT \ X=89 \ GOTO 210
130 P=A(1) \ I=A(2) \ N=A(3) \ Y=A(4) \ R=A(5) \ GOSUB 340
140 IF X THEN 150 ELSE 120
150 PRINT FNA$(17,8),%$C15F2,P \ PRINT FNA$(17,9),%15F2,I,"%"
160 PRINT FNA$(17,10),%12I,N \ IF INT(Y)<>Y THEN GOSUB 380
170 PRINT FNA$(17,11),%12I,Y, \ IF M THEN PRINT " Yrs.,",M," Mos." ELSE PRINT
180 PRINT FNA$(17,12),%$C15F2,R
190 PRINT FNA$(0,13) \ PRINT "Any more calculations? (Y-N): ",
200 X$=INCHAR$(0) \ X=ASC(X$) \ IF X>96 THEN X=X-32
210 IF X=78 THEN 220 \ IF X<>89 THEN 200 \ N=0 \ GOSUB 250 \ GOTO 80
220 PRINT FNA$(17,13),".. .. .. .. .. .. ..", \ END
230 IF A(3)<0 AND A(4)<0 THEN 240 \ X=89 \ GOTO 210
240 A(3)=12 \ PRINT FNA$(17,10),A(3) \ N=N-1 \ GOTO 100
250 PRINT FNA$(0,7) \ FOR I=1 TO 7 \ PRINT  \ NEXT \ PRINT FNA$(0,7)
260 PRINT "Principal amount: " \ PRINT "Annual interest :"
270 PRINT "No. payments/yr.:" \ PRINT "Term in years   :"
280 PRINT "Regular payment :",FNA$(18,8), \ RETURN
290 X$=INCHAR$(0) \ X=ASC(X$) \ IF X=13 THEN 330
300 IF X<>95 THEN 320 \ X=LEN(A$) \ IF X>1 THEN PRINT CHR$(95),
310 IF X>1 THEN A$=A$(1,X-1) \ GOTO 290
320 IF X<46 OR X>57 OR X=47 THEN 290 \ A$=A$+X$ \ PRINT X$, \ GOTO 290
330 IF LEN(A$)=1 THEN X=-1 ELSE X=VAL(A$) \ A$="0" \ RETURN
340 ON X GOTO 360,390,450,370,350
350 R=((I/100)*P/N)/(1-1/((I/100)/N+1)^(N*Y)) \ RETURN
360 P=R*N*(1-1/((I/100)/N+1)^(N*Y))/(I/100) \ RETURN
370 Y=-(LOG(1-(P*(I/100))/(N*R))/(LOG(1+I/100/N)*N))
380 M=INT(Y*12+.5) \ Y=INT(M/12) \ M=M-Y*12 \ RETURN
390 I=.1 \ I2=0 \ PRINT FNA$(18,9),"WORKING ... Stand by",FNA$(0,16),
400 R1=(I*P/N)/(1-1/((I/N+1)^(N*Y)))
410 R1=INT(R1*100+.5)/100 \ I3=ABS(I-I2)/2 \ I2=I
420 IF R1=R THEN 440 \ IF R1>R THEN I=I-I3 ELSE I=I+I3 \ GOTO 400
440 I=((INT((I*1000)*100+.5))/100)/1000 \ I=I*100 \ RETURN
450 PRINT FNA$(18,10),"This function NOT SUPPORTED",FNA$(0,16),
460 X=0 \ RETURN
470 DEF FNA$(X,Y)
480 IF Y<16 THEN 500 \ X$=CHR$(27)+CHR$(1)+CHR$(X)
490 PRINT X$, \ Y=CALL(49462) \ FILL 21841,0 \ X$="" \ GOTO 510
500 X$=CHR$(27)+CHR$(1)+CHR$(X)+CHR$(27)+CHR$(2)+CHR$(Y)
510 RETURN X$ \ FNEND
```

SUPERPACK
BY Bob Henderson

Recently I read a nice program by Rinaldo Prisco in Microcomputing (January 1981). Mr. Prisco's program compresses a North Star Basic Program by removing all the spaces (20H). While I have several programs that do the samething, the fascinating thing about such a program is that;
(1) it was in machine language so it was fast!, and
(2) having the source code allows you to modify it to your own special needs.

I have recently begun to learn machine language programming so this program was a motivation to begin expanding and adding features. What I found was that you can develop fairly complex programs by adding desired features once at a time and testing them as separate sub-programs.

First I wanted the program to remove superfluous REM statements. However, as I found out the hard way, removing all REM's can be a disaster. I know, a good programer never references a GOTO or GOSUB to a REM statement, but somehow I always find myself doing it. So the only REM statements this program can remove are ones not referenced. Naturally we eliminate the actual remarks and only leave the REM itself. (i.e. 95 H)

The other thing I wanted to have this program do was to concatenate statements. Not only does this allow a considerable reduction in the size of many programs but it also allows a somewhat faster operation. There is a program out in Pasic that does this but it can take up to 30 minutes on a big program. An assembly language program would be much faster.

The program simply goes through the original listing and every line number that is referenced is stored consecutively in TABLE. No attempt is made to eliminate duplications nor to order the table.

CON'T FROM PAGE 15: - SUPERPACK

This table is used by the routines which check for referenced REM statements, and for allowable concatenations. In spite of the theoretical inefficiency of building an over-large table and of requiring too many comparisions, the speed of the program is such that the effect is inconsequential.

Please remember if you concatenate statements larger than the allowable line length in your version of Basic, you will not be able to list or edit it. The length in the accompaning program is 96 but this can be changed.

The program is highly (probably excessively) documented, but this is essential for a novice like myself. Also I do not claim any awards for eligence. Fortunately machine language programs are so fast that even inefficient coding is very adequate.

To run the program:
(1) Load your program
(2) Type BYE to enter DOS
(3) Type GO SUPERPAK

The program is self prompting and will offer you two choices. First you will be asked if "you want to remove REM's." If you answer no, the program will remove spaces but not bother the REM's. If you answer Yes, you will then be asked if you wish "To concatenate statements." If you do, SUPERPAK will concatenate all statements to a maximum line length of 96 characters, except that referenced lines will not be moved.

When the program is complete, SUPERPAK will return to BASIC so you can run or save your program. I find that a 60 block, heavily remarked program, will compact to 37 blocks in about 2 seconds.

One word of warning. The program you are compacting must be complete. If you have an incomplete statement in your program:

```
(e.g.  100 GOTO
       200 PRINT "END"
       300 END
```

SuperPak will have unpredictable results.

I was greatly added in this new project by two very superior aids. The first was the XL-8080 Assembler System (Software Development Systems, P.O. Box 805 Mesa, Arizona, 85202).

Another indispensible aid is the Dynamic Debugging System (Computer Mart of New Jersey 501 Route 27, Iselin, NJ 08830). This is the finest system for debugging machine language programs I have seen. It allows you to review the program a step at a time while seeing all registers and flags. Great!

I hope the program will be useful to you.

Sincerely,
Bob Henderson
Bob Henderson

```
0000                  100 *        SUPERPAK          *
0000                  110 *
0000                  120 *   Public Domain Program by Bob Henderson 1981   *
0000                  130 *
0000                  140 * Written for SOL-20 with NORTH STAR DOS     *
0000                  150 *
0000                  160 * A program to use a line# reference table to allow
0000                  170 * us to delete all REM statements, EXCEPT those that
0000                  180 * are referenced by a GOTO, THEN, GOSUB, etc.
0000                  190 * and to optionally concatenate statements into
0000                  200 * multiple statement lines.
```

```
0000             210 *
0000             220 *
0000             230 * * * *   EQUATE TABLE   * * * *
0000             240 *
0000             250 VDMBAS EQU 0CC00H       TOP OF SCREEN
0000             260 BASIC EQU 2A04H        WARM RE-ENTRY TO BASIC
0000             270 HOWLONG EQU 96          MAX. LENGTH OF CONCAT. LINE
0000             280 SCREEN EQU 0CC80H      TWO LINES DOWN FROM TOP
0000             290 CLSCN  EQU 0C0D5H       SOLOS CLEAR SCREEN
0000             300 KEYBOARD  EQU 0FCH     SOL KEYBOARD INPUT
0000             310 STATUS EQU 0FAH         SOL KEYBOARD STATUS
0000             320 STARTBASIC  EQU  5E1EH     START OF BASIC FOR 5.2S
0000             330 *                          CHANGE STARTBASIC FOR OTHER
0000             340 *                          VERSIONS.
0000             350 TABLETOP  EQU 0B00H      Top of table area where we
0000             360 *                         store the ref. line#'s.
0000             370 *
0000             380 *
0000             390                           ORG 0H   I have located this version of
0000             400 *                   SUPERPAK below BASIC where I have
0000             410 *                   memory, but you can locate it
0000             420 *                   any place above your program also.
0000 CD D5 C0    430 CLEAR  CALL CLSCN     SOLOS CLEAR SCREEN
0003 21 00 CC    440        LXI H,VDMBAS
0006 36 20       450        MVI M," "            COVER CURSOR
0008 21 80 CC    460 BEGIN LXI H,SCREEN * PUT UP SIGN ON MESSAGES & CHOICES
000B 01 03 03    470       LXI B,MESS1
000E CD 25 01    480       CALL MSGO
0011 21 C4 CC    490       LXI H,SCREEN+68
0014 01 3C 03    500       LXI B,MESS12
0017 CD 25 01    510       CALL MSGO
001A 21 00 CD    520       LXI H,SCREEN+128
001D 01 70 03    530       LXI B,MESS2
0020 CD 25 01    540       CALL MSGO
0023 21 40 CD    550       LXI H,SCREEN+192
0026 01 AD 03    560       LXI B,MESS3
0029 CD 25 01    570       CALL MSGO
002C 21 80 CD    580       LXI H,SCREEN+256
002F 01 E9 03    590       LXI B,MESS4
0032 CD 25 01    600       CALL MSGO
0035 21 C0 CD    610       LXI H,SCREEN+320
0038 01 2A 04    620       LXI B,MESS5
003B CD 25 01    630       CALL MSGO
003E 21 00 CE    640       LXI H,SCREEN+384
0041 01 62 04    650       LXI B,MESS6
0044 CD 25 01    660       CALL MSGO
0047 21 80 CE    670       LXI H,SCREEN+512
004A 01 8A 04    680       LXI B,MESS7
004D CD 25 01    690       CALL MSGO
0050 CD 53 01    700       CALL DATAREADY        NEW INPUT?
0053 DB FC       710 KBRD  IN KEYBOARD  GET USER INPUT ABOUT REMOVAL OF REM
0055 FE 59       720       CPI "Y"
0057 CA 62 00    730       JZ GOOD
005A FE 4E       740       CPI "N"
005C CA 62 00    750       JZ GOOD
005F C3 53 00    760       JMP KBRD                ACCEPT ONLY "Y" OR "N"
0062 CD 47 01    770 GOOD  CALL TYPE ——    STORE "Y" OR "N"
0065 3A 00 03    780       LDA REMOVE        DO WE WANT TO REMOVE REMs?
0068 FE 59       790       CPI "Y"
006A C2 98 00    800       JNZ START    NO, THEN START PROGRAM
006D 21 C0 CE    810       LXI H,SCREEN+576     YES, ASK ABOUT
0070 01 D8 04    820       LXI B,MESS14           CONCATENATION.
0073 CD 25 01    830       CALL MSGO
0076 CD 53 01    840       CALL DATAREADY        NEW INPUT?
0079 DB FC       850 KBRD2 IN KEYBOARD
007B FE 59       860       CPI "Y"
007D CA 88 00    860       JZ GOOD2
0080 FE 4E       860       CPI "N"
0082 CA 88 00    860       JZ GOOD2    BE SURE Y OR N
0085 C3 79 00    870       JMP KBRD2
```

CON'T FROM PAGE 16: - SUPERPACK

```
ØØ88 32 Ø1 Ø3          88Ø GOOD2  STA CAT     STORE THE ANSWER
ØØ8B FE 59             89Ø        CPI "Y"
ØØ8D CC 5C Ø1          89Ø        CZ YES
ØØ9Ø FE 4E             89Ø        CPI "N"
ØØ92 CC 63 Ø1          89Ø        CZ NO
ØØ95 C3 76 Ø1          9ØØ        JMP REFLINE BUILD THE TABLE OF REF. LINE #'S
ØØ98                   91Ø *
ØØ98 11 1E 5E          92Ø START  LXI D,STARTBASIC     START OF PRGM. FOR BASIC 5.2S
ØØ9B 21 1E 5E          93Ø        LXI H,STARTBASIC     SET BOTH POINTERS TO BEGINNING
ØØ9E                   94Ø *                      OF BASIC PROGRAM
ØØ9E 1A                95Ø NEW LDAX D                  GET FIRST BYTE
ØØ9F 77                96Ø     MOV M,A         PUT IN "NEW LINE"
ØØAØ FE Ø1             97Ø     CPI 1           END OF FILE?
ØØA2 CA ØC Ø2          98Ø     JZ WANTCAT      CHECK FOR CONCATENATION
ØØA5 E5                99Ø     PUSH H        . SAVE ADDR. OF LENGTH BYTE
ØØA6 47                1ØØØ    MOV B,A
ØØA7 23                1Ø1Ø SKIP INX  H            SKIP OVER LINE #
ØØA8 13                1Ø2Ø      INX D
ØØA9 1A                1Ø3Ø      LDAX D
ØØAA 77                1Ø4Ø      MOV M,A
ØØAB 23                1Ø5Ø      INX H
ØØAC 13                1Ø6Ø      INX D
ØØAD 1A                1Ø7Ø      LDAX D
ØØAE 77                1Ø8Ø      MOV M,A
ØØAF 23                1Ø9Ø HIT INX H    CHECK FOR SPECIAL BYTES
ØØBØ 13                11ØØ     INX D
ØØB1 1A                111Ø     LDAX D
ØØB2 77                112Ø     MOV M,A
ØØB3 FE 9A             113Ø     CPI 9AH                 * NEXT WILL BE LINE#
ØØB5 CA A7 ØØ          114Ø     JZ SKIP
ØØB8 FE ØD             115Ø     CPI ØDH                 * END OF LINE
ØØBA CA FD ØØ          116Ø     JZ UPDATE
ØØBD FE 5C             117Ø     CPI 5CH                 * BACKLASH?
ØØBF CC ØA Ø1          118Ø     CZ  BACK                * YES, CHANGE FLAG BYTE

ØØC2 FE 8F             119Ø     CPI 8FH                 * REM
ØØC4 CA E2 ØØ          12ØØ     JZ REM
ØØC7 FE 22             121Ø     CPI 22H                 * QUOTE
ØØC9 CA D6 ØØ          122Ø     JZ QUOTE
ØØCC FE 2Ø             123Ø     CPI 2ØH                 * SPACE
ØØCE C2 AF ØØ          124Ø     JNZ HIT                 * EAT THE SPACE!
ØØD1 2B                125Ø     DCX H
ØØD2 Ø5                126Ø     DCR B
ØØD3 C3 AF ØØ          127Ø     JMP HIT
ØØD6                   128Ø QUOTE EQU $       DO NOT DISTRUB ITEMS IN QUOTES
ØØD6 23                129Ø      INX H
ØØD7 13                13ØØ      INX D
ØØD8 1A                131Ø      LDAX D
ØØD9 77                132Ø      MOV M,A
ØØDA FE 22             133Ø      CPI 22H          SECOND QUOTE MARK?
ØØDC C2 D6 ØØ          134Ø      JNZ QUOTE
ØØDF C3 AF ØØ          135Ø      JMP HIT
ØØE2 3A ØØ Ø3          136Ø REM  LDA REMOVE      LET'S SEE IF WE DELT. REM'S
ØØE5 FE 59             137Ø     CPI "Y"
ØØE7 C2 3B Ø1          138Ø     JNZ REMARK     NO, LEAVE THEM IN
ØØEA 13                139Ø     INX D          YES. LET'S RUN THRU LINE INCR. DE
ØØEB 1A                14ØØ     LDAX D            GET FOR COMPARE
ØØEC Ø5                141Ø     DCR B             REMEMBER TO REDUCE # OF BYTES
ØØED FE ØD             142Ø     CPI ØDH       END OF LINE ?
ØØEF C2 E2 ØØ          143Ø     JNZ REM       NO. KEEP ON
ØØF2 23                144Ø     INX H        TO PLACE FOR <CR>
ØØF3 CD 1Ø Ø1          145Ø     CALL CHK1    CHECK FOR BACKSLASH AT END OF LINE
ØØF6 36 ØD             146Ø     MVI M,ØDH
ØØF8 C3 1A Ø1          147Ø     JMP CHECK2    WAS THERE A BACKSLASH USED?
ØØFB Ø6 Ø5             148Ø REM2   MVI B,Ø5   CHANGE #OF BYTES FOR MIN. LINE
ØØFD 23                149Ø UPDATE INX H     MOVE HL AND DE UP ONE MORE
ØØFE AF                15ØØ        XRA A          ZERO A
ØØFF 32 Ø2 Ø3          151Ø        STA BACKSLASH    RESET
Ø1Ø2 13                152Ø        INX D   SO THEY ARE AT START OF NEXT LINE
Ø1Ø3 E3                153Ø        XTHL       NOW THE CRITICAL STEP. THE ADDR. OF THE
Ø1Ø4                   154Ø *          #OF BYTES/LINE IS ON THE STACK. EXCHANGE
```

```
Ø1Ø4                   155Ø *          THAT WITH HL (ie HL now points to # byte)
Ø1Ø4 78                156Ø            MOV A,B NOW STORE NEW # OF BYTES
Ø1Ø5 77                157Ø            MOV M,A
Ø1Ø6 E1                158Ø            POP H            FINALLY HL POINTS TO START OF NEW LINE
Ø1Ø7 C3 9E ØØ          159Ø            JMP NEW
Ø1ØA                   16ØØ *
Ø1ØA 3E Ø1             161Ø BACK  MVI A,Ø1      FLAG FOR BACKSLASH IS Ø1
Ø1ØC 32 Ø2 Ø3          162Ø       STA BACKSLASH
Ø1ØF C9                163Ø       RET
Ø11Ø                   164Ø *
Ø11Ø 3A Ø2 Ø3          165Ø CHK1  LDA BACKSLASH
Ø113 FE ØØ             166Ø       CPI ØH
Ø115 C8                167Ø       RZ  NO BLASKSLASH USED
Ø116 Ø5                168Ø       DCR B       YES,  SO ADJUST POINTERS
Ø117 2B                169Ø       DCX H
Ø118 2B                17ØØ       DCX H
Ø119 C9                171Ø       RET
Ø11A                   172Ø *
Ø11A 3A Ø2 Ø3          173Ø CHECK2  LDA BACKSLASH   BACKSLASH FLAG
Ø11D FE ØØ             174Ø         CPI Ø
Ø11F CA B9 Ø1          175Ø         JZ READTABLE     NO BACKSLASH
Ø122 C3 FD ØØ          176Ø         JMP UPDATE     BACKSLASH
Ø125                   177Ø *
Ø125 ØA                178Ø MSGO  LDAX B       TO DISPLAY MESSAGES
Ø126 FE FF             179Ø       CPI ØFFH
Ø128 C8                18ØØ       RZ
Ø129 77                181Ø       MOV M,A
Ø12A Ø3                182Ø       INX B
Ø12B 23                183Ø       INX H
Ø12C C3 25 Ø1          184Ø       JMP MSGO
Ø12F                   185Ø *
Ø12F ØA                186Ø MSGOR  LDAX B    SAME AS MSGO BUT IN
Ø13Ø FE FF             187Ø        CPI ØFFH       REVERSE VIDEO
Ø132 C8                188Ø        RZ
Ø133 F6 8Ø             189Ø        ORI 8ØH
Ø135 77                19ØØ        MOV M,A
Ø136 Ø3                191Ø        INX B
Ø137 23                192Ø        INX H
Ø138 C3 2F Ø1          193Ø        JMP MSGOR
Ø13B                   194Ø *
Ø13B 23                195Ø REMARK INX H      DO NOT CHANGE REMS SINCE WE
Ø13C 13                196Ø        INX D      ARE NOT DELETING THEM
Ø13D 1A                197Ø        LDAX D
Ø13E 77                198Ø        MOV M,A
Ø13F FE ØD             199Ø        CPI ØDH
Ø141 C2 3B Ø1          2ØØØ        JNZ REMARK
Ø144 C3 FD ØØ          2Ø1Ø        JMP UPDATE
Ø147                   2Ø2Ø *
Ø147                   2Ø3Ø TYPE EQU $            STORE 'Y' OR 'N' for
Ø147 32 ØØ Ø3          2Ø4Ø      STA REMOVE   REMOVAL OF REM STATEMENTS
Ø14A FE 59             2Ø5Ø      CPI "Y"
Ø14C CA 5C Ø1          2Ø6Ø      JZ YES       PRINT YES ON SCREEN
Ø14F CD 63 Ø1          2Ø7Ø      CALL NO
Ø152 C9                2Ø8Ø      RET
Ø153 DB FA             2Ø9Ø DATAREADY  IN STATUS  GET SERIAL STATUS
Ø155 2F                21ØØ            CMA    INVERT FOR SOL
Ø156 E6 Ø1             211Ø            ANI 1
Ø158 CØ                212Ø            RNZ
Ø159 C3 53 Ø1          213Ø            JMP DATAREADY
Ø15C                   214Ø *
Ø15C Ø1 D1 Ø4          215Ø YES  LXI B,MESS1Ø
Ø15F CD 25 Ø1          216Ø      CALL MSGO
Ø162 C9                217Ø      RET
Ø163                   218Ø *
Ø163 Ø1 D5 Ø4          219Ø NO         LXI B,MESS11
Ø166 CD 25 Ø1          22ØØ            CALL MSGO
Ø169 C9                221Ø            RET
Ø16A                   222Ø *
Ø16A Ø1 A8 Ø4          223Ø DONE       LXI B,MESS8
Ø16D 21 CØ CE          224Ø            LXI H,SCREEN+576
Ø17Ø CD 2F Ø1          225Ø            CALL MSGOR      RESERVSE VIDEO
```

CON'T FROM PAGE 17: - SUPERPACK

```
Ø173 C3 Ø4 2A      2260          JMP BASIC  RE-ENTER BASIC
Ø176               227Ø *
Ø176               228Ø REFLINE   EQU $   START OF ROUTINE TO BUILD A
Ø176               229Ø *                 TABLE OF REFERENCED LINE#'S
Ø176 11 ØØ ØB      23ØØ                   LXI D,TABLETOP  USE DE TO POSITION IN TABLE
Ø179 21 1E 5E      231Ø                   LXI H,STARTBASIC       HL WILL SCAN PROGRAM
Ø17C 7E            232Ø NEW1  MOV A,M       FIRST BYTE OF PRGM
Ø17D FE Ø1         233Ø       CPI 1                END OF FILE?
Ø17F CA B1 Ø1      234Ø       JZ  FINISHED  YES. RETURN TO BASIC
Ø182 23            235Ø SKIPOVER  INX H   SINCE WE ARE AT START OF LINE
Ø183 23            236Ø           INX H   WE WILL SKIPOVER OVER LINE #
Ø184 23            237Ø CHECK     INX H   NOW SEE IF BYTE
Ø185 7E            238Ø           MOV A,M  IS GOING TO REF ANOTHER LINE#
Ø186 FE 9A         239Ø           CPI 9AH  IS NEXT BYTE LINE#?
Ø188 CA A2 Ø1      24ØØ           JZ FOUND  OK LET'S STORE IT.
Ø18B FE ØD         241Ø           CPI ØDH  END OF LINE?
Ø18D CA AD Ø1      242Ø           JZ NEWLINE   START AGAIN
Ø19Ø FE 22         243Ø           CPI 22H  A QUOTE?
Ø192 CA 98 Ø1      244Ø           JZ QUOTE1
Ø195 C3 84 Ø1      245Ø           JMP CHECK
Ø198 23            246Ø QUOTE1    INX H
Ø199 7E            247Ø           MOV A,M
Ø19A FE 22         248Ø           CPI 22H  KEEP GOING UNTIL NEW QUOTE
Ø19C C2 98 Ø1      249Ø           JNZ QUOTE1
Ø19F CA 84 Ø1      25ØØ           JZ CHECK
Ø1A2               251Ø *
Ø1A2 23            252Ø FOUND     INX H   POINT TO LSB
Ø1A3 7E            253Ø           MOV A,M  PUT LSB IN A
Ø1A4 12            254Ø           STAX D  STORE LSB IN TABLE
Ø1A5 23            255Ø           INX H
Ø1A6 13            256Ø           INX D
Ø1A7 7E            257Ø           MOV A,M  MOST SIGNIFICANT BIT
Ø1A8 12            258Ø           STAX D   STORE IT
Ø1A9 13            259Ø           INX D   GET READY FOR NEXT TIME
Ø1AA C3 84 Ø1      26ØØ           JMP CHECK   NEXT!
Ø1AD               261Ø *
Ø1AD 23            262Ø NEWLINE  INX H   GET READY FOR NEW LINE
Ø1AE C3 7C Ø1      263Ø          JMP NEW1
Ø1B1 3E FF         264Ø FINISHED  MVI A,ØFFH   MARK END OF TABLE WITH FFFFH
Ø1B3 12            265Ø           STAX D
Ø1B4 13            265Ø           INX D
Ø1B5 12            265Ø           STAX D
Ø1B6 C3 98 ØØ      266Ø           JMP START     NOW START PACK SECTION OF PROGRAM
Ø1B9               267Ø *
Ø1B9               268Ø READTABLE  EQU $           FIND IF LINE# IS IN TABLE
Ø1B9 2B            269Ø            DCX H  THIS IS THE REM
Ø1BA 2B            27ØØ            DCX H
Ø1BB 7E            27ØØ            MOV A,M
Ø1BC 32 FF Ø2      27ØØ            STA MSB   MSB OF CURRENT LINE#
Ø1BF 2B            271Ø            DCX H
Ø1CØ 7E            271Ø            MOV A,M
Ø1C1 32 FE Ø2      271Ø            STA LSB
Ø1C4 2B            272Ø            DCX H
Ø1C5 2B            272Ø            DCX H     POINTS TO LAST ØDH
Ø1C6 C5            273Ø            PUSH B
Ø1C7 E5            273Ø            PUSH H            SAVE REG.
Ø1C8               274Ø *
Ø1C8 21 ØØ ØB      275Ø            LXI H,TABLETOP ADDRESS OF START OF TABLE
Ø1CB 3A FE Ø2      276Ø CHECKLSB  LDA LSB         GET CURRENT LINE# LSB
Ø1CE BE            277Ø           CMP M                   SAME AS IN TABLE?
Ø1CF CA DD Ø1      278Ø           JZ  CHECKMSB            HAVE MATCH NOW SEE MSB
Ø1D2 3E FF         279Ø           MVI A,ØFFH              CHECK FOR END OF TABLE
Ø1D4 BE            28ØØ           CMP M
Ø1D5 CA E9 Ø1      281Ø           JZ ENDOFTABLE
Ø1D8 23            282Ø           INX H
Ø1D9 23            282Ø           INX H           OTHERWISE GOTO NEXT ENTRY
Ø1DA C3 CB Ø1      283Ø           JMP CHECKLSB
Ø1DD               284Ø *
Ø1DD 3A FF Ø2      285Ø CHECKMSB  LDA MSB         GET MSB
Ø1EØ 23            286Ø           INX H
```

```
Ø1E1 BE            287Ø          CMP M                   MATCH?
Ø1E2 CA F6 Ø1      288Ø          JZ INTABLE
Ø1E5 23            289Ø          INX H
Ø1E6 C3 CB Ø1      29ØØ          JMP CHECKLSB            SO TRY ANOTHER
Ø1E9               291Ø *
Ø1E9 23            292Ø ENDOFTABLE  INX H         CHECK FOR DOUBLE FF
Ø1EA BE            293Ø             CMP M STILL ØFFH?
Ø1EB 23            294Ø             INX H
Ø1EC C2 CB Ø1      295Ø             JNZ CHECKLSB  NO.
Ø1EF               296Ø *
Ø1EF E1            297Ø NOTINTABLE        POP H
Ø1FØ C1            297Ø                   POP B
Ø1F1 23            298Ø                   INX H
Ø1F2 13            298Ø                   INX D   READY FOR NEW LINE
Ø1F3 C3 9E ØØ      299Ø                   JMP NEW         THIS LINE IS DEAD, GET ANOTHER
Ø1F6               3ØØØ *
Ø1F6 E1            3Ø1Ø INTABLE  POP H
Ø1F7 C1            3Ø1Ø          POP B    RESTORE REG.
Ø1F8 23            3Ø2Ø          INX H
Ø1F9 23            3Ø2Ø          INX H
Ø1FA 23            3Ø2Ø          INX H
Ø1FB 23            3Ø2Ø          INX H  RESTORE TO END
Ø1FC 23            3Ø3Ø          INX H    MOVE TO START OF NEW LINE
Ø1FD AF            3Ø4Ø          XRA A
Ø1FE 32 Ø2 Ø3      3Ø4Ø          STA BACKSLASH   RESET
Ø2Ø1 C3 FB ØØ      3Ø5Ø          JMP REM2 SAVE THIS LINE AS MIM. LINE WITH REM
Ø2Ø4               3Ø6Ø *
Ø2Ø4               3Ø7Ø *          START OF THE CONCATENATING SECTION   *
Ø2Ø4 3A ØØ Ø3      3Ø8Ø            LDA REMOVE     HAVE REMS BEEN REMOVED?
Ø2Ø7 FE 59         3Ø9Ø            CPI "Y"
Ø2Ø9 C2 6A Ø1      31ØØ            JNZ DONE        NO. THEN RETURN TO BASIC
Ø2ØC               311Ø *
Ø2ØC 3A Ø1 Ø3      312Ø WANTCAT  LDA CAT  ARE WE SUPPOSE TO CONCATENATE?
Ø2ØF FE 59         313Ø          CPI "Y"
Ø211 CA 17 Ø2      313Ø          JZ STARTCAT
Ø214 C3 6A Ø1      313Ø          JMP DONE     IF NOT THEN
Ø217               314Ø *
Ø217               315Ø STARTCAT  EQU $
Ø217 11 1E 5E      316Ø            LXI D,STARTBASIC
Ø21A 21 1E 5E      317Ø            LXI H,STARTBASIC
Ø21D 1A            318Ø NEW2   LDAX D
Ø21E 77            319Ø        MOV M,A
Ø21F FE Ø1         32ØØ        CPI 1
Ø221 CA EB Ø2      321Ø        JZ  QUIT
Ø224 47            322Ø        MOV B,A
Ø225 3A FC Ø2      323Ø        LDA NEWLENGTH    GET PRESENT LENGTH
Ø228 FE ØØ         324Ø        CPI Ø
Ø22A C2 38 Ø2      325Ø        JNZ CONCAT   IF  Ø THEN WE ARE AT START OF LINE
Ø22D E5            326Ø        PUSH H
Ø22E 78            327Ø        MOV A,B
Ø22F 32 FD Ø2      327Ø        STA OLDLENGTH
Ø232 32 FC Ø2      327Ø        STA NEWLENGTH
Ø235 C3 6F Ø2      328Ø        JMP JUMPOVER
Ø238               329Ø *
Ø238 8Ø            33ØØ CONCAT    ADD B            ADD LENGTH OF PRESENT TO NEW
Ø239 3D            331Ø           DCR A
Ø23A 3D            331Ø           DCR A
Ø23B 3D            331Ø           DCR A   REDUCE FOR FIRST OF LINE
Ø23C FE 6Ø         332Ø           CPI HOWLONG  LINES UP TO 96 (max. line length)
Ø23E D2 66 Ø2      333Ø           JNC DONOT      COMBINED LINE LENGTH TOO LONG
Ø241 32 FC Ø2      334Ø           STA NEWLENGTH
Ø244 47            335Ø           MOV B,A    B=LENGTH OF NEW COMBINED LINE
Ø245 13            336Ø           INX D
Ø246 1A            336Ø           LDAX D
Ø247 32 FE Ø2      336Ø           STA LSB          GET LSB & STORE IT
Ø24A 13            337Ø           INX D
Ø24B 1A            337Ø           LDAX D
Ø24C 32 FF Ø2      337Ø           STA MSB          GET MSB & STORE IT
Ø24F 1B            338Ø           DCX D
Ø25Ø 1B            338Ø           DCX D   RESTORE POINTERS
Ø251 CD BC Ø2      339Ø           CALL INTABLE2  SEE IF WE CAN CONCATENATE
```

CON'T FROM PAGE 18: - SUPERPACK

```
Ø254                3400 *                RET WITH A=FF IF NOT IN  OR A=Ø IF IN
Ø254 FE ØØ          3410                  CPI Ø         IN TABLE?
Ø256 CA 66 Ø2       3420                  JZ DONOT      YES, SO DO NOT CONCAT.
Ø259 2B             3430                  DCX H
Ø25A 36 5C          3430                  MVI M,5CH   REPLACE WITH BACKSLASH
Ø25C 3A FC Ø2       3440                  LDA NEWLENGTH      USE BECAUSE WE WILL CONCAT
Ø25F 32 FD Ø2       3450                  STA OLDLENGTH      UPDATE IT
Ø262 47             3460                  MOV B,A
Ø263 C3 7A Ø2       3470                  JMP JUMPOVER2      AVOID LINE #
Ø266                3480 *
Ø266 3A FD Ø2       3490 DONOT   LDA OLDLENGTH
Ø269 47             3500         MOV B,A        CORRECT LENGTH
Ø26A 2B             3510         DCX H
Ø26B 1B             3510         DCX D      CORRECT POINTERS FOR NEXT LINE
Ø26C C3 A7 Ø2       3520         JMP UPDATE2
Ø26F                3530 *
Ø26F 23             3540 JUMPOVER INX  H    SKIP LINE #
Ø270 13             3550          INX D
Ø271 1A             3560          LDAX D
Ø272 77             3570          MOV M,A
Ø273 23             3580          INX H
Ø274 13             3590          INX D
Ø275 1A             3600          LDAX D
Ø276 77             3610          MOV M,A
Ø277 C3 7C Ø2       3620          JMP COMPARE
Ø27A                3630 *
Ø27A 13             3640 JUMPOVER2 INX D   JUMPOVER OVER LINE# WITH DE
Ø27B 13             3650           INX D   BUT ELIMINATE FROM NEW CODE
Ø27C 23             3660 COMPARE INX H
Ø27D 13             3670         INX D
Ø27E 1A             3680         LDAX D
Ø27F 77             3690         MOV M,A
Ø28Ø FE 9A          3700         CPI 9AH
Ø282 CA 6F Ø2       3710         JZ JUMPOVER
Ø285 FE ØD          3720         CPI ØDH
Ø287 CA B7 Ø2       3730         JZ NEWLINE2       ALL DONE.
Ø28A FE 8F          3740         CPI 8FH   REM
Ø28C CA A3 Ø2       3750         JZ REMS
Ø28F FE 22          3760         CPI 22H
Ø291 CA 97 Ø2       3770         JZ QUOTE2
Ø294 C3 7C Ø2       3780         JMP COMPARE
Ø297                3790 *
Ø297                3800 QUOTE2 EQU $
Ø297 23             3810        INX H
Ø298 13             3820        INX D
Ø299 1A             3830        LDAX D
Ø29A 77             3840        MOV M,A
Ø29B FE 22          3850        CPI 22H
Ø29D C2 97 Ø2       3860        JNZ QUOTE2
Ø2AØ C3 7C Ø2       3870        JMP COMPARE
Ø2A3                3880 *
Ø2A3 13             3890 REMS  INX D       WE NEVER CONCAT. TO A REM
Ø2A4 23             3900       INX H
Ø2A5 1A             3910       LDAX D
Ø2A6 77             3920       MOV M,A              SO PUT THE ØD ON AND UPDATE
Ø2A7 23             3930 UPDATE2 INX H
Ø2A8 AF             3940         XRA A              ZERO A
Ø2A9 32 FC Ø2       3950         STA NEWLENGTH      ZERO LENGTH
Ø2AC 32 FD Ø2       3960         STA OLDLENGTH      ZERO LENGTH
Ø2AF 13             3970         INX D
Ø2BØ E3             3980         XTHL
Ø2B1 78             3990         MOV A,B
Ø2B2 77             4000         MOV M,A
Ø2B3 E1             4010         POP H
Ø2B4 C3 1D Ø2       4020         JMP NEW2
Ø2B7                4030 *
Ø2B7 23             4040 NEWLINE2 INX H
Ø2B8 13             4040          INX D      AFTER ØD PREPARE FOR NEW
Ø2B9 C3 1D Ø2       4050          JMP NEW2
Ø2BC                4060 *
Ø2BC E5             4070 INTABLE2   PUSH H         SAVE HL
Ø2BD 21 ØØ ØB       4080            LXI H,TABLETOP  ADD. OF START OF REF. TABLE
Ø2CØ 3A FE Ø2       4090 CHECKLSB2 LDA LSB         GET CURRENT LINE# LSB
Ø2C3 BE             4100           CMP M                   SAME AS IN TABLE?
Ø2C4 CA D2 Ø2       4110           JZ  CHECKMSB2   HAVE MATCH NOW SEE MSB
Ø2C7 3E FF          4120           MVI A,ØFFH   CHECK FOR END OF TABLE
Ø2C9 BE             4130           CMP M
Ø2CA CA DE Ø2       4140           JZ ENDTABLE
Ø2CD 23             4150           INX H
Ø2CE 23             4150           INX H           OTHERWISE GOTO NEXT ENTRY
Ø2CF C3 CØ Ø2       4160           JMP CHECKLSB2
Ø2D2                4170 *
Ø2D2 3A FF Ø2       4180 CHECKMSB2  LDA MSB                GET MSB
Ø2D5 23             4190            INX H
Ø2D6 BE             4200            CMP M                  MATCH?
Ø2D7 CA E8 Ø2       4210            JZ MATCH
Ø2DA 23             4220            INX H                  NO
Ø2DB C3 CØ Ø2       4230            JMP CHECKLSB2  SO TRY ANOTHER
Ø2DE                4240 *
Ø2DE 23             4250 ENDTABLE INX H    CHECK FOR DOUBLE FF
Ø2DF BE             4260          CMP M           STILL FF?
Ø2EØ 23             4270          INX H
Ø2E1 C2 CØ Ø2       4280          JNZ CHECKLSB2
Ø2E4                4290 *
Ø2E4 E1             4300 NOTHERE   POP H           RESTORE HL
Ø2E5 3E FF          4310           MVI A,ØFFH
Ø2E7 C9             4320           RET
Ø2E8                4330 *
Ø2E8 E1             4340 MATCH     POP H
Ø2E9 AF             4350           XRA A   A=Ø MEANS IN TABLE
Ø2EA C9             4360           RET
Ø2EB                4370 *
Ø2EB 3A FD Ø2       4380 QUIT      LDA OLDLENGTH
Ø2EE E3             4390           XTHL    EXCHANGE ADD. OF LEN.BYTE FROM STACK
Ø2EF 77             4400           MOV M,A
Ø2FØ 21 ØØ CF       4410           LXI H,SCREEN+64Ø
Ø2F3 Ø1 A8 Ø4       4420           LXI B,MESS8
Ø2F6 CD 2F Ø1       4430           CALL MSGOR
Ø2F9 C3 Ø4 2A       4440           JMP BASIC   RETURN TO BASIC
Ø2FC                4450 *
Ø2FC                4460 * * *    STORAGE BYTES    * * *
Ø2FC                4470 *
Ø2FC ØØ             4480 NEWLENGTH DB Ø    LENGTH OF CURRENT LINE
Ø2FD ØØ             4490 OLDLENGTH  DB Ø  LINE BEFORE ADDING NEW
Ø2FE ØØ             4500 LSB DB Ø
Ø2FF ØØ             4510 MSB DB Ø
Ø3ØØ ØØ             4520 REMOVE  DB Ø      STOREAGE FOR REMOVE REMS?
Ø3Ø1 ØØ             4530 CAT  DB Ø         STORE SHALL WE CONCATENATE?
Ø3Ø2 ØØ             4540 BACKSLASH  DB Ø   BACKSLASH IN CURRENT LINE?
Ø3Ø3                4550 *
Ø3Ø3                4560 * * *    MESSAGES    * * *
Ø3Ø3                4570 *
Ø3Ø3 2Ø 2Ø 2Ø       4580 MESS1  "   ***** North Star SUPERPAK program for BASIC 5.2S ****"
Ø33B FF             4590        DB   ØFFH
Ø33C 2A 2A 2A       4600 MESS12    "****   by Bob Henderson - 1981 (Public Domain) ****"
Ø36F FF             4610           DB ØFFH
Ø37Ø 54 68 69       4620 MESS2  "This program will pack North Star Basic programs by deleting"
Ø3AC FF             4630        DB   ØFFH
Ø3AD 61 6C 6C       4640 MESS3  "all spaces. In addition the program will also allow you to "
Ø3E8 FF             4650        DB   ØFFH
Ø3E9 65 69 74       4660 MESS4  "either remove or not remove REM statements. Only non-referrence
Ø429 FF             4670        DB   ØFFH
Ø42A 52 45 4D       4680 MESS5  "REMs will be removed. If they are removed, you can then"
Ø461 FF             4690        DB   ØFFH
Ø462 6F 7Ø 74       4700 MESS6  "optionally concatenate lines (max.=96)."
Ø489 FF             4710        DB   ØFFH
Ø48A 44 6F 2Ø       4720 MESS7  "Do you want to remove REMs ? "
Ø4A7 FF             4730        DB   ØFFH
Ø4A8 2Ø 41 4C       4740 MESS8  " ALL DONE.  I will now return to BASIC. "
Ø4DØ FF             4750        DB   ØFFH
```

CON'T FROM PAGE 19: - SUPERPACK

```
Ø4D1 59 45 53           4760 MESS1Ø  "YES"
Ø4D4 FF                 477Ø         DB   ØFFH
Ø4D5 4E 4F              478Ø MESS11  "NO"
Ø4D7 FF                 479Ø         DB   ØFFH
Ø4D8 44 6F 2Ø           48ØØ MESS14    "Do you want to concatenate statements ? "
Ø5ØØ FF                 481Ø           DB ØFFH
PROGRAM IS 1281 BYTES LONG (6) WITH Ø ERRORS DETECTED.
> SYMBOL LISTING (Y=YES,N=NO)?

BACK.......Ø1ØA  BACKSLASH..Ø3Ø2  BASIC......2AØ4  BEGIN......ØØØ8
CAT........Ø3Ø1  CHECK......Ø184  CHECK2.....Ø11A  CHECKLSB...Ø1CB
CHECKLSB2..Ø2CØ  CHECKMSB...Ø1DD  CHECKMSB2..Ø2D2  CHK1.......Ø11Ø
CLEAR......ØØØØ  CLSCN......CØD5  COMPARE....Ø27C  CONCAT.....Ø238
DATAREADY..Ø153  DONE.......Ø16A  DONOT......Ø266  ENDOFTABLE.Ø1E9
ENDTABLE...Ø2DE  FINISHED...Ø1B1  FOUND......Ø1A2  GOOD.......ØØ62
GOOD2......ØØ88  HIT........ØØAF  HOWLONG....ØØ6Ø  INTABLE....Ø1F6
INTABLE2...Ø2BC  JUMPOVER...Ø26F  JUMPOVER2..Ø27A  KBRD.......ØØ53
KBRD2......ØØ79  KEYBOARD...ØØFC  LSB........Ø2FE  MATCH......Ø2E8
MESS1......Ø3Ø3  MESS1Ø.....Ø4D1  MESS11.....Ø4D5  MESS12.....Ø33C
MESS14.....Ø4D8  MESS2......Ø37Ø  MESS3......Ø3AD  MESS4......Ø3E9
MESS5......Ø42A  MESS6......Ø462  MESS7......Ø48A  MESS8......Ø4A8
MSB........Ø2FF  MSGO.......Ø125  MSGOR......Ø12F  NEW........ØØ9E
NEW1.......Ø17C  NEW2.......Ø21D  NEWLENGTH..Ø2FC  NEWLINE....Ø1AD
NEWLINE2...Ø2B7  NO.........Ø163  NOTHERE....Ø2E4  NOTINTABLE.Ø1EF
OLDLENGTH..Ø2FD  PSW........ØØØ6  QUIT.......Ø2EB  QUOTE......ØØD6
QUOTE1.....Ø198  QUOTE2.....Ø297  READTABLE..Ø1B9  REFLINE....Ø176
REM........ØØE2  REM2.......ØØFB  REMARK.....Ø13B  REMOVE.....Ø3ØØ
REMS.......Ø2A3  SCREEN.....CC8Ø  SKIP.......ØØA7  SKIPOVER...Ø182
SP.........ØØØ6  START......ØØ98  STARTBASIC.5E1E  STARTCAT...Ø217
STATUS.....ØØFA  TABLETOP...ØBØØ  TYPE.......Ø147  UPDATE.....ØØFD
UPDATE2....Ø2A7  VDMBAS.....CCØØ  WANTCAT....Ø2ØC  YES........Ø15C
*
```

INFO WANTED:

I recently purchased a California Computer Systems Model 2422 Floppy Disk Controller for use with my SOL. I have, however, been unable to get it working.

I thought perhaps some other SOL owner may have successfully married the 2422 with their SOL and would be willing to share the information.

I would certainly appreciate any assistance that PROTEUS may offer.

Thank you.

Sincerely,

H. T. Hoffman, Ph.D.

HOWARD T. HOFFMAN
5545 STRESEMANN STREET
SAN DIEGO, CALIFORNIA 92122

TELEPHONE (714) 453-3891

Communication Services
Computerized Business Systems
Software and Hardware Consulting
Phone: (203) 522-5252

• P.O.Box 333
• Hartford, CT 06101

• 166 Collins St. #101
• Hartford, CT 06105

DEAR COMPUTER DEALER:

DISKETTE CONVERSION SERVICE

Here is an opportunity for your company to provide a valuable service to your customers, earn their goodwill and get paid well for the time spent. Note that this is also an excellent solution to CP/M software developers looking to open up new markets.

We'll convert standard 8" CP/M format (the so-called IBM 3740 single density format) to 5" North Star CP/M format (both single or double density) and vice versa. This conversion service retails $21.00 for the first converted diskette and $15.00 each additional. As a computer dealer, however, you pay only $11.00 for the first converted diskette and $8.00 each additional converted diskette in the same order. You supply the media or we bill you for it at cost. Our turnaround time is typically 24 hours. To quality for the dealer discount, write your order on your letterhead. Dealer order must be $27.00 minimum or 3 converted diskettes per order. Of course, you are at liberty to set your own retail markup. The original diskettes will be shipped together with the converted diskettes and confidentiality is assured. Volume discounts are available for replicas---Please write or call for details.

July 24, 1981

Stan Sokolow
1690 Woodside Rd., Suite 219
Redwood City, Ca. 94061

Dear Stan,

Ever since I joined PROTEUS the beginning of this year, I have been pleased with the information in the PROTEUS NEWS. However I would like some help with the following problems. Does someone have a transfer routine for transferring programs from a PT cassette tape to a North Star mini-floppy diskette and vice versa? I would like to have my 32K block of memory stay continuously at OH or 2000H whichever is better. Also is there a disk utility pack and some word-processing programs available without expanding my system to CPM? Finally what would be involved for a future addition of an 8080/8085 assembler to my SOL/Northstar equipment?

Any information would be greatly appreciated.

Sincerely yours,

Dale Funke

Dale Funke
10102 Crescent Moon
Houston, Tx. 77064

PATCHES FOR TREK-80, QUBIC, 8080-CHESS
BY Tom Cardoso

Having recently converted SOLOS to operate at F000 hex, I became frustrated that some of the great P.T. software would no longer operate, since most of it uses direct screen memory calls. An effort was made to convert some of the good stuff, and the following are patches to allow QUBIC, 8080-CHESS and TREK-80 run with screen memory addressed starting at FC00 hex. As time permits, I will also try to get TARGT running and if successful, will forward the patches to PROTEUS for publication.

The patches for QUBIC, 8080-CHESS and TREK-80 allow the programs to run with relocated SOLOS, but extensive testing of the modified programs has not been done. If any bugs are evident as a result of the move, please forward them to me.

To modify, load program into memory starting at 0 hex, make patches, then save the modified version. If done under CP/M, load at 100 hex, add 100 hex to the addresses given, change the bytes indicated and save 8 pages for QUBIC, 24 pages for ·CHESS, and 33 pages for TREK-80.

PATCHES FOR TREK-80

Address	Was	Should be	Address	Was	Should be
0032	CD	FD	08E8	CE	FE
005E	CE	FE	0975	CC	FC
0067	CC	FC	0990	CC	FC
006E	D0	00	0993	CC	FC
0156	CF	FF	0996	CD	FD
0161	CD	FD	09C8	CC	FC
0166	CD	FD	09E7	CC	FC
016E	CD	FD	09FB	CC	FC
017B	CD	FD	0A10	CC	FC
0184	CC	FC	0A2D	CC	FC
018A	CF	FF	0B1A	CC	FC
01B2	C0	F0	0B54	CC	FC
022B	CC	FC	0B6B	CC	FC
031E	CC	FC	0BBB	CC	FC
0337	CC	FC	0C59	CC	FC
0350	CD	FD	0CDB	CC	FC
0365	CE	FE	0CE0	D0	00
039F	CF	FF	0D14	CF	FF
03C6	CF	FF	0D20	CF	FF
03D6	CD	FD	0D56	CC	FC
03DF	CD	FD	0E11	CE	FE
03E7	CD	FD	0E3B	CE	FE
03F0	CD	FD	0E65	CC	FC
03FC	CC	FC	0E77	CC	FC
04A3	CC	FC	0E81	CC	FC
04A7	CC	FC	0E97	CC	FC
05B3	CC	FC	0E9B	CC	FC
0608	CC	FC	0EA2	CF	FF
0648	CC	FC	0EAE	CC	FC
064C	CC	FC	0EE2	CD	FD
069E	CD	FD	0EE6	CD	FD
06E1	CF	FF	0EE9	30	70
0706	CC	FC	0EEE	CC	FC
0722	CF	FF	0F10	CD	FD
07A7	CC	FC	0F1A	CF	FF
07CE	CC	FC	0F23	CC	FC
07F1	CC	FC	0F3C	CD	FD
07F7	CF	FF	0F54	CD	FD
0818	CC	FC	1FF4	CD	FD
081E	CC	FC	**178F**	**C0**	**F0**

NOTE: For some reason, the "L" command to leave TREK-80 is non-functional after the patches are installed. That doesn't pose much of a problem, but if anyone has ideas on fixing that, let me know.

PATCHES FOR QUBIC

Address	Was	Should be	Address	Was	Should be
021F	CC	FC	05C1	CC	FC
0225	D0	00	05DC	CF	FF
0245	CF	FF	0615	CF	FF
033A	CC	FC	0628	CF	FF
034D	CF	FF	0634	CC	FC
05BB	CC	FC	0637	CD	FD

PATCHES FOR 8080-CHESS V.1.0 (MOD 0)

Address	Was	Should be	Address	Was	Should be
0023	CC	FC	068C	CC	FC
0060	CF	FF	0695	CC	FC
0069	CF	FF	06C9	CF	FF
00C4	CF	FF	06E0	CF	FF
00DB	CF	FF	10EA	CC	FC
0146	CF	FF	1112	CF	FF
021F	CF	FF	11D5	CC	FC
0495	CF	FF	11DB	CC	FC
04BE	CC	FC	11E4	CD	FD
04EC	CF	FF	125C	CC	FC
055B	CC	FC	126E	CC	FC
0560	CC	FC			

Additionally, add the following hex code at 16CF which overwrites the checksum error message:

```
16CF: 0B 50 6C 65 61 73 65 20 68 69 74 20 60 52 45 54 55 52 4E 27
```

*** GOOD LUCK ***

Tom Cardoso
1469 Farington Dr.
Naperville, Ill. 60540

INFO WANTED ON NEW SOL BOARD:

I WOULD LIKE TO REQUEST SOME ASSISTANCE CONCERNING THE FOLLOWING ITEM. I HOPE THAT ANOTHER MEMBER MIGHT BE ABLE TO COME TO MY AID (PROVIDE SUCCOR) WITH SOME OR ALL OF THESE.
#1 SOL COMPUTER BOARD (C) 1978 , REVISION B ASSEMBLY #110200: PC #110201, REVISION B. WHAT ARE THE VALUES OF THE FOLLOWING COMPONENTS ?????
C 75 (NEAR U113)?
C 78 (NEAR U 25)?
R 91 (BETWEEN U 75 AND U 76)?
R160 (NEAR U 39)?
U117 (NEXT TO DOUBLE ROW OF DIP SWITCHES)?
THOSE ITEMS ARE PREVENTING ME FROM COMPLETING MY COMPUTER!!!

VICTOR D. BENNIGHT
COMPUTER SCIENCES CORPORATION
1101 SAN ANTONIO ROAD
MOUNTAIN VIEW, CA 94043

ED NOTE: THIS IS THE PC BOARD FOR THE REDESIGNED SOL WHICH WAS NOT ON THE MARKET WHEN PTC WENT OUT OF BUSINESS. WE HAVE NO DOCUMENTATION. PERHAPS SOMEONE ELSE HAS FIGURED IT OUT.

CUSTOM I/O PRINTER DRIVER
BY Wayne Wilson

Enclosed is an enhanced version my print driver that appeared in the January/February Proteus. The driver is for an ASL2 Multiwriter which has a Diablo Hytype 1 printer assembly. This version of the driver still provides handshaking as before, but now collects "white space" and moves the print head directly to the position of the next character to be printed in one big "tab" rather than by printing individual spaces across the page. This driver is implemented as part of my CP/M BIOS and by using the Iobyte function, switching from 10 CPI to 12 CPI is done via a STAT command from the keyboard.

```
****************************************************************
*                           ASL2                               *
* Custom I/O printer driver for Multiwriter printer with Hytypel*
* assembly.  Handshake for 1200 baud and collected spaces.     *
*                          (10 cpi)                            *
****************************************************************
coul1   XRA     A
        STA     TAB
SEROUT  PUSH    H       ;H/L WILL BE USED
        LHLD    RIGHT   ;GET THE CURRENT COUNT
        MOV     A,C
        CPI     20H     ;CHECK FOR A SPACE
        JZ      INC10   ;HOLD CARRIAGE, COLLECT SPACES
        LDA     SPACES  ;CHECK TO SEE IF
                        ;LAST PRINTABLE CHARACTER
        CPI     01      ;WAS A SPACE.
        JNZ     SOUT    ;IF NOT SEND CHARACTER OUT
                        ;
                        ;AFTER COLLECTING SPACES
                        ;SEND OUT ONE BIG TAB.
                        ;
        MOV     A,C
        STA     TEMPRY  ;SAVE THE LAST PRINTABLE
                        ;CHARACTER IN AFTER SPACES
        CALL    BFLAG   ;SEND OUT THE BUFFER FLAG
        CALL    WAIT    ;LOOK FOR ANSWER-BAK, THEN MOVE
        MVI     C,01CH  ;LEAD IN CODE FOR A MOVE COMMAND
        CALL    SOUT
        MOV     C,H     ;NOW SEND OUT DIRECTION
        CALL    SOUT    ;    OF MOTION
        MOV     C,L     ;AND HOW FAR TO MOVE THE
        CALL    SOUT    ;    CARRIAGE
        XRA     A       ;TIME TO RESET
        STA     RIGHT   ;THE TAB REGISTER
        STA     RIGHT+1
        STA     SPACES  ;AND LOWER THE FLAG
        LDA     TEMPRY  ;GET CHARACTER BEING STORED
        MOV     C,A     ;PUT IT IN C REG.
SOUT    CALL    colpt   ;SEND A CHARACTER OUT
        LDA     CHRCNT  ;GET CHARACTER COUNT
        INR     A       ;BUMP THE COUNT
        STA     CHRCNT  ;STORE NEW COUNT
        CPI     60      ;SENT 60 CHARACTERS?
        JZ      WAIT    ;IF SO STAND-BY
        CPI     55      ;IF 60, IS IT 55?
        JNZ     BACK    ;IF NOT, GET MORE
BFLAG   MVI C,  05H     ;IF=55, SEND OUT BUFFER MARKER
        CALL    colpt   ;TO DIABLO
BACK    LDA     SPACES  ;GET FLAG
        CPI     1       ;CHECK FOR SPACES
        JZ      RTRN    ;GET MORE MOTION DATA
POPS    POP     H
RTRN    RET             ;BACK FOR MORE
WAIT    IN      0F8H    ;GET STATUS
        ANI     040H    ;MASK FOR SOL SER. DATA READY
        JZ      WAIT    ;LOOP ROUND TILL READY
        IN      0F9H    ;GET DATA
```

```
        ANI     07FH
        CPI     07FH    ;IS IT DIABLO ANS-BAK
        JNZ     WAIT    ;IF NO TRY AGAIN
        XRA     A       ;AFTER ANS-BAK, RESET
        STA     CHRCNT  ;RESET THE COUNTER
        JMP     BACK    ;RETURN FOR MORE
;
;SPACES COLLECTED HERE BASED ON 60 INCREMENTS PER
;INCH (5 increments per char. for 12 cpi)
;     (6 increments per char. for 10 cpi)
;
INC10   MVI     A,1     ;RAISE THE 'SPACES' FLAG
        STA     SPACES
        LDA     TAB     ;FIND OUT IF WE ARE
        CPI     1       ;PRINTING 12 cpi
        JZ      INC12   ;IF YES, SWITCH TO 12
        MOV     A,L     ;INCREMENT MOVE COUNT BY 6
        ADI     6       ;6/60TH INCREMENT FOR 10 CPI
        MOV     L,A
        JMP     MASK    ;CHECK THE HIGH ORDER BIT
INC12   MOV     A,L     ;INCREMENT MOVE COUNT BY 5
        ADI     5       ;5/60TH INCREMENT FOR 12 CPI
        MOV     L,A
MASK    ANI     80H     ;MASK FOR HIGH ORDER BIT
        CPI     80H     ;WHEN COUNT OVER 127
        JZ      HROUT   ;GO INCREMENT "H" REG.
        SHLD    RIGHT   ;IF NOT, STORE THE COUNT
        JMP     POPS    ;GO BACK FOR MORE STUFF
HROUT   MOV     A,L     ;IF COUNT OVER 127
        SUI     128     ;BY HOW MUCH
        MOV     L,A     ;PUT REMAINDER BACK IN L
        INR     H       ;COUNT EXTRA IN 'H'
        SHLD    RIGHT
        JMP     POPS    ;THEN GO BACK FOR MORE
;
CHRCNT  DB      00H
SPACES  DB      00H     ;1 IF PREVIOUS CHARACTER WAS A SPACE
RIGHT   DW      0000H   ;HERE WE COUNT ALL THE COLLECTED SPACES
TEMPRY  DB      00H
TAB     DB      00H     ;12 cpi IF FLAG IS '1'

****************************************************************
*                                                              *
*    Set flag for printing at 12 cpi on ASL2 Multiwriter.      *
*                                                              *
****************************************************************
coul2   MVI     A,1     ;Set up flag for
        STA     TAB     ;printing 12 cpi.
        JMP     SEROUT  ;To main print routine
****************************************************************
```

I am currently working on implementing vertical motion in a similar fashion and bi-directional printing. Ultimately, I plan to burn the whole driver into an Eprom and have it on board as firmware in my system. I will forward the complete driver as it develops, in the event it may be of interest to others. As I said last time, if anyone out there is familiar with this printer and has a better way of doing this...I'm all ears!

The money order enclosed is to cover the cost of Helios Diskette H-6, as listed in the November/December 1980 Proteus News at $35.00.

In passing, I have used the SSM VB3 memory-mapped 80x24 video board in my SOL, as well as the SD Sales VDB-8024 I/O mapped board and the Imsai VIO 80x24 memory mapped board. All of these boards worked well and without any particular problem. The VB3 board with its on board Z-80 running at 4MHZ presents a very fast display. For word processing applications, an I/O mapped board such as the VDB-8024 seems preferable, as it does not occupy system Ram. My biggest objection to the Imsai VIO board is the fact that the on board Prom resides at F000 and is in

CON'T FROM PAGE 22: - DRIVER FOR PRINTER

conflict with my Solos 56K system. It does work well with Solos at C000 of course. By using a coaxial relay, both the 80x24 and VDM 64x16 displays are readily available.

The SOL is alive and well...Keep up the good work.

Your truly

Wayne Wilson

8022 - 117th Street
Delta, B.C. V4C 6A9
CANADA
May 31, 1981

USING SOL AS A TERMINAL

Robert Heerdink wrote to Proteus asking this question: "Can anyone recommend the easiest, cheapest way for me to use my 32K Sol-20, no-disk system to access the Source or any other network using a modem? Since I have Solos, can I just wire the serial port to the phone and use the TE command?"

The Sol was designed to be useful as a dumb terminal as well as a smart computer. To successfully link your keyboard and video screen to the host computer at the other end of the telephone line, you must do three things.

First, you need a device called a "modem", which stands for "modulator/demodulator". This changes the computer's digital (on/off) signals into tones that can be transmitted over ordinary telephone circuits. The Sol's serial communications port is designed to act like the serial port on a terminal with the RS-232 communications standard.

The modem will have a socket that will take a cable from the Sol serial connector. The connectors needed on the cable are known as "DB-25P" connectors. Not all 25 pins on the connectors need to be wired with the cable. Only the following pins are essential for simple operation as a terminal: 2, 3, 7, 20. Note that pin 20 on the Sol, which is supposed to signal that the terminal is "ready", is always in the Ready state when the Sol is on, even if the Sol is not in terminal mode. But sure to get Sol ready before you engage the computer.

Radio Shack sells an inexpensive modem that accepts a cable from the Sol and will plug directly into a modular phone jack. Some modems connect to the phone line indirectly (acoustically), using a small speaker and microphone in rubber cups that the telephone handset fits into. The direct-connect modems are preferred.

Second, you must set the switches on the modem and inside the Sol to make it behave the way that the host computer expects. Some computer systems will tolerate a variety of terminals and so they can adapt themselves to the setting of some switches. However, it is best to inquire from the network just what parameters their computer expects. The variable parameters are:

Baud (transmission rate, that is, the number of bits per second sent over the line, ordinarily no more than 300 baud);

Parity--an error detection bit, may be Even, Odd, or omitted altogether;

Number of stop bits--bits sent at the end of each character, usually one or two.

Number of data bits--usually 7 or 8;

Duplex mode--signifies whether the characters received by the host computer are to be echoed back to the terminal for display on the screen (full duplex), or not (half duplex).

Consult the Sol manual appendix AVII for tables showing the S3 and S4 switch settings. Note that some early manuals had a typographical error in the data word length settings for switch S4. They should be:

Data word length	S4-2	S4-3
8 bits	off	off
7 bits	on	off
6 bits	off	on
5 bits	on	on

Third, you must put the Sol into terminal mode using the TE command and call-up the computer. When the computer answers the phone, you will hear a carrier tone. Then you should switch the modem from voice to data transmission, and you should be in business. Consult the modem manual for specifics of establishing the link. Be sure that they Sol is not in Local mode (the local key on the keyboard should be off) or no data will be sent to the serial port; instead it is simply echoed on the screen.

Some computers expect you to send a certain character, such as carriage return, as the first thing you send. This allows their equipment to analyze your transmission parameters and adapt. When a successful link is established, you should receive a sign-on message from the host.

Some keys on the Sol may not behave as you expect. The gray keys should be okay, except that the DEL key doesn't send the ASCII backspace. In fact, Sol doesn't have a key that sends backspace. You have to press the H key while holding down the control key. The Control-H character is the standard backspace, but Solos's Terminal command doesn't process it as a backspace. Instead, it puts a curved arrow on the screen. This may confuse you, but after a while you will get used to it. To improve the TE command, you can write an assembly language version of your own that does what you want. After all, the Terminal mode of the Sol is simply a program.

The new Proteus manual on the Sol (see this issue's story on Encyclopedia Processor Technica) contains a chapter detailing the use of serial communications in the Sol. Watch for our announcement of the manual in the next issue.

LETTERS TO THE EDITOR:

...ON RUNNING PTC SOFTWARE UNDER CP/M,

27 JUNE 1981

DEAR STAN,

ENCLOSED IS AN ASSEMBLED LISTING OF A QUICK-AND-DIRTY METHOD I USE TO STORE AND RUN PT GAME SOFTWARE ON MY VISTA V-200 CP/M SYSTEM - THIS SHOULD BE OF USE TO BRUCE DILLER OR ANYONE ELSE WHO HAS WONDERED HOW TO RUN PT GAMES FROM DISK. TO SUMMARIZE, THE 47 BYTE ROUTINE IS LOADED INTO MEMORY FROM 100H TO 12FH UNDER CP/M, THE PT SOFTWARE THEN "TACKED ON" TO THE ROUTINE BY LOADING IT AT 130H UNDER SOLOS, CP/M REBOOTED, AND THE FILE FROM 100 TO THE END OF THE PT SOFTWARE SAVED AS A .COM FILE. AFTER THIS IS DONE, THE PT SOFTWARE CAN BE RUN FROM DISK BY TYPING THE NAME. NOTE THAT THIS IS NOT A CONVERSION TO RUN UNDER CP/M PER SE - WHEN YOU EXIT A PT PROGRAM RUN IN THIS MANNER, YOU RETURN TO SOLOS JUST AS IF YOU HAD LOADED FROM CASSETTE. RETURN TO CP/M REQUIRES A COLD START.

I USED THIS METHOD TO LOAD EC BASIC FROM DISK - STILL HAD TO STORE AND LOAD BASIC PROGRAMS FROM CASSETTE, BUT THE 8 SECOND LOAD TIME FOR BASIC MADE ME FEEL BETTER. THEN I GOT THE TAD

CON'T FROM PAGE 23 - PTC SOFTWARE UNDER CP/M

MODIFICATION TO EC BASIC>... IF ANYONE HASN'T SEEN THE WRITE-UPS IN PREVIOUS PROTEUS ISSUES, I CAN HIGHLY RECOMMEND THE TAD ENTERPRISES CONVERSION TO EC BASIC WHICH ALLOWS EC BASIC TO READ AND WRITE DATA TO CP/M DISKS.

SHORT HARDWARE REVIEW: THE SSM PB-1 PROM BURNER WORKS GREAT! YOU CAN BURN 2708'S AND 2716'S, AND IT HAS 4K WORTH OF EXTRA PROM SOCKETS ON IT THAT CAN BE ADDRESSED WHEREVER YOU LIKE. SOFTWARE IS INCLUDED WITH THE PB-1.

ON THE SUBJECT OF EPROMS, AN ERASER IS NECESSARY. BYTE MAGAZINE PUBLISHED AN ARTICLE A FEW MONTHS BACK DETAILING THE CONSTRUCTION OF AN EPROM ERASER USING A SYLVANIA FT6/BLB ULTRAVIOLET BULB. I BUILT SAME, AND IT WOULD NOT ERASE MY TEXAS INSTRUMENT 2516 PROM. BASICALLY THE REASON TURNED OUT TO BE THAT THE BLB-SERIES BULBS PUT OUT THE WRONG FLAVOR ULTRAVIOLET. THEIR OUTPUT IS CENTERED AROUND 360 NM, WHEREAS THE SPECS ON THE TI PROM CALLED FOR 259 NM. THE ONE IN BYTE MAGAZINE MAY HAVE WORKED WELL WITH OTHER PROMS, BUT IT DIDN'T CAUSE THE TI EPROM TO DROP A BIT. THE SOLUTION WAS TO TRADE THE BLB TUBE IN FOR A SYLVANIA GERMICIDAL BULB. THE 12 WATT GERMICIDAL BULB, STARTER, BALLAST, AND ASSORTED SOCKETS, SWITCH AND FUSE ALL FIT NICELY INTO THE BOTTOM OF A K-MART 8-TRACK STEREO TAPE CASE (PLASTIC, $2), AND WITH THE EPROM 1 INCH FROM THE TUBE, THEY ARE COMPLETELY ERASED IN 25 MINUTES. THE WHOLE DEVICE WAS BUILT FOR $26, WHICH IS ABOUT $50 LESS THAN THE COMMERCIAL ERASERS I'VE SEEN ADVERTISED. ONE NOTE OF CAUTION - THE GERMICIDAL UV TUBE EMITS SHORT WAVE ULTRAVIOLET WHICH WILL BURN YOUR SKIN OR EYES. FORTUNATELY SHORT WAVE UV WILL NOT PENETRATE GLASS, OR OPAQUE MATERIALS, SO IT IS EASY TO PROTECT AGAINST EXPOSURE. JUST DON'T LOOK AT THE TUBE TO SEE IF IT WORKS! ANY ELECTRICAL SUPPLY HOUSE SHOULD STOCK THE PARTS, AND PROVIDE YOU WITH A WIRING DIAGRAM .

NOTE TO JOE MAGUIRE: IN YOUR EXCELLENT SERIES ABOUT REPAIRING SOL'S, YOU MENTIONED THAT REPLACING IC'S WILL CURE MOST ILLS, EXCEPT PERHAPS DAMAGE DUE TO LIGHTNING - WELL, MY SOL GOT HIT BY LIGHTNING IN FLORIDA WHILE HOOKED UP TO A HAM RADIO DURING A BAD STORM (DUMB) - AND I WANTED TO REPORT THAT YOU CAN FIX THAT BY CHANGING IC'S. YOU JUST HAVE TO CHANGE ALOT MORE OF THEM...(EVERYTHING FROM THE PARALLEL PORT TO THE CPU - INCLUDING THE CPU.). JUST HAD TO THROW THAT IN!

BEST REGARDS,

Leigh Torgerson

J. LEIGH TORGERSON
22410 BARBACOA DRIVE
SAUGUS, CA 91350

```
*** BLOCK MOVE AND SET-UP TO RUN PT SOFTWARE UNDER CP/M ***
*
* THIS IS A SIMPLE ROUTINE TO ENABLE THE USE OF PT SOFTWARE
* WITH CP/M DISK SYSTEMS. ALL IT DOES IS ENABLE YOU TO STORE
* PT SOFTWARE ON DISK AND RUN THE SOFTWARE LIKE A .COM FILE
*
*                        NOTE!
*
* I DON'T WANT TO MISLEAD ANYONE - THIS DOES NOT INTERFACE
* PT SOFTWARE TO CP/M IN ANY WAY - IT JUST LETS YOU LOAD
* THINGS LIKE TREK80, ECBASIC, 8080 CHESS ETC. FROM DISK
* RATHER THAN FROM TAPE. AFTER THIS ROUTINE, CP/M HAS
* BEEN OVERWRITTEN, AND YOU MUST PERFORM A COLD-START
* FROM SOLOS TO GET BACK INTO OPERATING WITH CP/M.
*
* TO SAVE PT SOFTWARE AND RUN FROM CP/M, TWO BASIC REQUIREMENTS
* MUST BE MET:
```

```
                  *     1. SINCE CP/M LOADS THINGS FROM THE DISK INTO
                  *        MEMORY STARTING AT 0100H, PT SOFTWARE READ
                  *        IN FROM DISK MUST BE MOVED TO 0000H BEFORE
                  *        IT CAN BE RUN.
                  *     2. PT SOFTWARE EXPECTS THE STACK POINTER TO BE
                  *        SET UP AS WELL AS SOME SOLOS ENTRY POINTS
                  *        IN THE STACK AND THE HL REGISTERS.
                  *
                  * NOTICE THAT ONCE ITEM ONE HAS BEEN PERFORMED, THE CP/M
                  * SOFTWARE HAS BITTEN THE DUST, SINCE IT USES 0 TO 100 FOR
                  * VARIOUS BOOKKEEPING.
                  *
                  * THIS PROGRAM WILL TAKE CARE OF THE ABOVE ITEMS. FIRST,
                  * IT RELOCATES ITSELF TO AN AREA IN SOLOS USER RAM -
                  * (WATCH OUT IF YOU HAVE DRIVERS OR ANYTHING THERE..)
                  * THEN IT MOVES THE 20K BLOCK FROM 0130H UP TO ZERO,
                  * AND FINALLY SETS UP THE STACK POINTER, STACK AND HL REGISTER
                  * LIKE SOLOS WOULD HAVE IF YOU HAD PERFORMED AN XEQ PROGNAME.
                  *
                  * TO USE THIS ROUTINE, LOAD IT INTO MEMORY STARTING AT
                  * 100H USING DDT. NEXT, DO AN UPPER CASE-REPEAT RESET BACK
                  * INTO SOLOS. LOAD YOUR PT PROGRAM FROM TAPE USING A LOAD
                  * ADDRESS OF 0130H (I.E. >GET PROG 130). ONCE THE PROGRAM
                  * IS LOADED, CALCULATE HOW MANY 256 BYTE PAGES TO SAVE,
                  * BOOT CP/M AND TYPE IN SAVE XX PROG.COM. AFTER THIS IS DONE,
                  * YOU MAY EXECUTE THE PT PROGRAM BY TYPING IT'S NAME, LIKE
                  * ANY  COM FILE.
                  *
0100                        ORG     0100H
0100 111301       RELOC     LXI     D,MOVE          ;START OF MOVE PROGRAM
0103 2150C9                 LXI     H,0C950H        ;MOVE IT TO C950H
0106 1A           NXTBY     LDAX    D
0107 77                     MOV M,A
0108 23                     INX     H
0109 13                     INX     D
010A 7B                     MOV     A,E
010B FE30                   CPI     030H
010D C20601                 JNZ     NXTBY
0110 C350C9                 JMP     MOVE+BIAS
C83D =            BIAS      EQU     0C950H-0113H
                  *
0113 113001       MOVE      LXI     D,PROG          ;FILE TO BE MOVED
0116 210000                 LXI     H,0000          ;WANNA MOVE IT TO ZERO
0119 1A           NEXT      LDAX    D               ;STUFF AT DE INTO A REG.
011A 77                     MOV     M,A             ;PUT INTO LOCATION HL
011B 23                     INX     H               ;BUMP HL PAIR
011C 13                     INX     D               ;AS WELL AS DE
                  *
                  * NOW PUT MSB OF ADDRESS OF DATA BEING MOVED INTO A-REG
                  * AND COMPARE TO CHECK ADDRESS - AS IMPLEMENTED HERE,
                  * BLOCK FROM PROG TO 4FFFH WILL BE MOVED UP TO ZERO.
                  * (ABOUT 20K  BYTES - LONGER THAN ANY PT SOFTWARE I HAVE..)
                  *
011D 7A                     MOV     A,D
011E FE50                   CPI     050H
0120 C256C9                 JNZ     NEXT+BIAS
                  *
                  * NOW THAT THE PT PROGRAM HAS BEEN MOVED TO 0000H,
                  * SET UP SP, FIRST ENTRY ON STACK, AND HL TO MAKE
                  * PT PROGRAMS HAPPY..
                  *
0123 31FFCB                 LXI     SP,0CBFFH       ;SOLOS STACK POINTER
0126 2104C0                 LXI     H,0C004H        ;SOLOS RETURN TO SYS ENTRY POINT
0129 E5                     PUSH    H               ;NEEDS TO BE ON STACK.
012A 2100C0                 LXI     H,0C000H        ;BEGINNING OF SOLOS IN HL
012D C30000                 JMP     0000            ;NOW RUN PT PROGRAM!
0130 00           PROG      NOP                     ;PT PROGRAM STARTS HERE
                  END

A>
```

...ON THE HAZARDS OF REPAIRS

ALERT
BEWARE THE REPAIRMANS TRAP

In hopes that others may avoid the sort of financial booby-trap that I recently fell into, I would like to tell how I managed to do so. Just being aware that it can happen and taking a few precautions may save someone a rude surprise.

We have had our two Helios in operation since March of 1978. Except for two occasions, both involving head positioning lamp failures, no repairs were necessary during that time. We have often wondered if some sort of preventitive maintenance might be called for, so, although both Helios were operating, when the eject mechanism began acting up on one and the positioning servo on the other began making strange noises, we contacted the service shop that had replaced one of the burned-out lamps. We were satisfied that the owner was qualified; he worked for PerSci before he went into business for himself, and the earlier work he had done for us was satisfactory. For that work the charge had been $85.00, quite a bit but not excessive when you understand that at least a partial realignment is necessary on that repair. He told us that he would give both the PerScis a complete checkout, clean the heads, replace the pressure pads, and "..make them as good as new ". Those were his words.

I picked up the Helios three days later; the bill read:

```
(1) PerSci 277 S.N.
    Work Done:
    Full alignment:
    Attach Positioner Servo Grounds:
    Adjust index side 1.
      Parts replaced:    (2) pressure pads
                              @ $1.00 each            $2.00
      Labor:                 4 hours
                              @ $50.00 each         $200.00
(2) PerSci 277 S.N.
    Work Done:
    Full Alignment:
    Clean eject sensor side 1:
      Parts Replaced:    (2) pressure pads
                              @ $1.00 each            $2.00
                          (2) cones @ $12.00         $24.00
      Labor:                  4 hours
                              @ $50.00 each         $200.00
                                                    -------
                                            Total: $458.00
```

Notice that the figures do not add up to $458.00; I didn't when I paid the bill. Later I called it to their attention and they sent me a check for $24.00. But that inaccuracy is not the bone of my contention.

I was shocked at the amount; most people would be I think. At this point there is little one can do but complain. He has your many thousands of dollars worth of Helios while all you have is a sick feeling in the pit of your stomach and a lot less money. Unfortunately the worse is yet to be told. Within a week after the Helios were back in operation, the very same conditions for which they were taken for repairs had reoccurred. The eject mechanism on one some times takes half a dozen pushes before it works, and the Positioner Servo on the other occasionally does strange things, the same strange things as before.

When I paid the bill the first time, he assured me that his work was guaranteed for thirty days, and when I called him to report that the repairs had failed he did not hesitate to remind me of that. I returned the Helios with the positioner problem and repairs were completed within two days. I will return the other soon and I'm sure this time it too will be properly repaired.

So what am I crying about? First the size of the bill. Fifty dollars an hour is unreasonable and I simply do not believe it took eight hours for the job. I suspect that some of that time involves the machines simply running unattended. Probably the charges are made according to some schedule; so many hours for this and so many for that. I think thats unfair. I'm upset mostly because the machines were not properly repaired. One deserves more at those prices. I recieved very little for my money the first time around. Certainly not what one is entitled to for fifty dollars an hour. And when it is a shop that obviously can do better, it's down right dirty pool. I am convinced that they are not all that interested in working on Helios because when I inquired about a service policy he said they did not like to write them on Helios. He said the necessity of taking the PerScis out of the Helios cabinet for repairs often caused more problems.

How can one avoid such experiences? Two things for sure. Dont insist on hurried repairs unless its absolutely neccessary. It puts one in debt to the repairman for doing something special, and one must bear part of the blame for bad repairs caused by rushing. Most important, ask for a written estimate. If they are reluctant to give one, ask why. An experienced technician will have an idea of what the repairs will entail. Finally, dont assume that rates are the same as last year (or last week). Inflation has hit the cost of repairs very hard.

I am not going to mention the name of the shop involved in this case. It obviously was not his fault that I failed to take the steps I recommend. And some may not agree that my experience was so bad and that the charges were unreasonable. I cannot help feeling that I have been had.

Better luck to all;

Earl J. Dunham
941 N. Russell
La Habra, CA 90631

Ed Note: I' sure you'll find Ace Computer Inc. service's rates for SOL or HELIOS repair more reasonable.

Ray and Bob can repair your S-100 equipment and peripherals with special expertise in Processor Tech, NorthStar and Morrow products.

Call Ace Computer Inc. (415)283-6630 M T W / 24 Hr. answering machine, TH F S (9:30-5:30) personal answering service.

Computer programs stolen

BELMONT — Thieves avoided an alarm system and escaped with $1,500 in computer programs by breaking through a wall into a store late Friday or early Saturday, police reported today.

Belmont Police Lt. James Scales said the thieves broke into a small shop next door to Computerland, 1625 El Camino Real.

The burglars then broke through a sheetrock wall with a crowbar and entered Computerland through an eight-foot-square hole, Scales said.

The thieves left with about $1,500 worth of computer business programming, Scales said.

JUNE 29, 1981

REPRINTED FROM PENINSULA TIMES TRIBUNE NEWSPAPER (SAN FRANCISCO PENINSULA AREA)

EDITOR'S REMARK: I'VE HEARD OF SOFTWARE THEFT BEFORE, BUT THIS IS THE HARD WAY, I'D SAY.

...ON MICRO-COMPLEX'S DUAL PERSONALITY MODULE

STAN,

I THINK I HAVE COME UPON SOMETHING IMPORTANT FOR ALL PROTEUS MEMBERS TO BE AWARE OF. LAST SUNDAY I WENT TO THE ADVANCED COMPUTER PRODUCTS ANNUAL SWAP-MEET. THERE I MET BOB HOGG, FORMERLY WITH ORANGE COUNTY COMPUTER CENTER IN COSTA MESA WHICH WAS THE BIG PROCESSOR TECHNOLOGY DEALER IN THAT PART OF THE STATE. HE HAD SET UP A LITTLE STAND AND WAS SELLING ARTICLES OF HIS OWN DESIGN AND MANUFACTURE FOR SOL-HELIOS AND SOL-NORTH STAR AND SOL-HELIOS-NORTH STAR SYSTEMS.

THIS GUY REALLY KNOWS WHAT HE IS DOING, STAN. HE HAS SOL PERSONALITY MODULES THAT ARE SENSATIONAL. I BOUGHT THE HELIOS VERSION AND IT WORKS PERFECTLY. WHAT HE HAS DONE IS PUT TWO SWITCH SELECTABLE ROMS ON THE MODULE. ONE PUTS SOLOS AT C000 AND THE OTHER PUTS SOLOS AT F000. YOU NEED ONLY REACH BEHIND SOL AND FLICK THE SWITCH TO GO FROM ONE TO THE OTHER. YOU DONT HAVE TO GET UP FROM YOUR CHAIR TO DO IT. BUT HE HAS DONE MUCH MORE THAN THAT. HE HAS MADE REAL IMPROVEMENTS IN SOLOS. HE HAS ADDED 20 COMMANDS WHICH I WAS SCEPTICAL ABOUT AT FIRST BUT WHICH I HAVE FOUND ARE REALLY IMPROVEMENTS. I AM INCLUDING COPIES OF SOME OF THE MANUAL THAT COMES WITH THE MODULE SO YOU CAN SEE FOR YOURSELF. AND THE PRICE IS RIGHT, ONLY $95.00.

THE INCREASED MEMORY CAPABILITY MAKES IT POSSIBLE TO RUN SOME OF THE CP/M APPLICATION SOFTWARE FORMERLY UNUSABLE FOR US BECAUSE OF OUR BEING LIMITED TO 48K. FOR THE SOL-NORTH STAR OWNER THE GAINS USING HIS DPM (DUAL PERSONALITY MODULE) MAY BE EVEN MORE IMPRESSIVE. YOU CAN READ ABOUT THAT IN THE ENCLOSED COPY OF THE MANUAL. AND IF ANYONE IS THINKING OF ADDING NORTH STAR TO THEIR SYSTEM, HIS CONTROLLER IS WORTH LOOKING INTO. IMAGINE, 819,000 BYTES PER DISK. THE COMBINATION OF HELIOS AND NORTH STAR WITH BOTH DOS IN MEMORY AT THE SAME TIME IS MIND BOGGLING.

BUT THERE IS MORE, AND IT SHOULD BE ESPECIALLY INTERESTING TO YOU AND THE REST OF US HOPING TO MAKE PTDOS WHAT IT SHOULD BE. WHILE AT O.C.C. HE WAS WORKING QUITE CLOSELY WITH PROC. TECH. IN THEIR DEVELOPMENTAL WORK JUST BEFORE THE COLLAPSE (HE HAS A NUMBER OF THE PROTOTYPE MINI-DISKS THEY WERE NOT ABLE TO BRING TO COMPLETION). STAN, HE SAYS THAT PTDOS COULD BE THE BEST DOS EVER MADE FOR SMALL COMPUTERS, BUT DUE TO THE DEATH OF THE DEVELOPEMENTAL ENGINEER BEFORE ITS COMPLETION, AND BECAUSE WHOEVER FINISHED IT DIDNT UNDERSTAND WHAT ITS FULL POTENTIAL WAS, IT IS NOT, AS IT NOW EXISTS, ANYWHERE NEAR WHAT IT COULD BE. HE TOLD ME HE IS CERTAIN THAT THE POTENTIAL FOR GREAT INCREASES IN SPEED AND POWER EXISTS WITHIN PTDOS, HE IS VERY ANXIOUS TO WORK WITH SOMEONE ON THE SOURCE CODE. READ HIS COMMENTS IN THE COPIES I HAVE ENCLOSED. I THINK THOSE WHO WILL BE WORKING ON THE SOURCE CODE FOR PTDOS, WHEN YOU FINALLY GET IT, WOULD DO WELL TO TALK TO THIS GUY.

I AM NOT TRYING TO SELL MR. HOGGS MERCHANDISE FOR HIM, BUT I HAVE BEEN FOOLING WITH SOL FOR NEARLY FOUR YEARS NOW AND I KNOW WHEN SOMEONE KNOWS WHAT THEY ARE TALKING ABOUT IN REGARDS TO SOL, AND THIS FELLOW IMPRESSES ME AS THE MOST KNOWLEDGABLE AND CAPABLE I HAVE MET LATELY. SINCE I AM INTERESTED, AS I'M SURE YOU ARE, IN SEEING SOL EXPLOITED TO ITS FULLEST AND SINCE THE MORE PEOPLE WE CAN GET INVOLVED WITH SOL THE MORE CHANCE THERE IS THAT PEOPLE LIKE BOB HOGG WILL INVEST THEIR TIME AND TALENT IN DEVELOPING WHAT WE NEED, I HOPE YOU WILL PUT THIS IN PROTEUS SO THAT EVERYONE WILL HAVE THE OPPORTUNITY TO TAKE ADVANTAGE OF THIS DEVELOPMENT FOR SOL. MR. HOGGS WORK ON A 24X80 DISPLAY AND ON THE HARD DISK AND OTHER IDEAS HE HAS FOR SOL ARE EXCITING AND I HOPE WE CAN ENCOURAGE HIM TO CONTINUE.

I HOPE THAT YOU WILL TALK TO BOB HOGG ABOUT THE WORK ON PTDOS. THE FULL DEVELOPEMENT OF HELIOS COULD PUT US BACK IN THE BIG LEAGUES IN SMALL COMPUTERS.

EARL J. DUNHAM
941 N. RUSSELL
LA HABRA, CA 90631
PH: (213) 697-7238

SINCERELY

Earl

EARL J. DUNHAM

MICRO COMPLEX
25651 MINOS STREET
MISSION VIEJO, CA. 92691
(714) 770-2168
15 February 1981

Micro Complex DUAL PERSONALITY MODULE for SOL Computers

A new product from Micro Complex can greatly enhance the application and versatility of the Processor Technology, Corp. SOL Computer. A single switch on the DPM (Dual Personality Module) provides switched operation of the SOL at either 0C000H or 0F000H monitor addresses. Also the switch provides switching between either of two on-board PROM monitor programs. Operation in the SOL is really quite impressive. As an example, switching to the 0F000H mode the screen display is moved to 0FC00H, the jump start moves to 0F000H ,SOL system memory moves to 0F800H, and if your are using a 64K memory board, you automatically get the memory back at 0C000H where your old SOLOS was addressed. This means that now you can run 60K of user memory space instead of 48K as in the originally configured SOL. This is very significant since many of the new appplication programs now available on the market require at least 56K of user memory space.

The DPM is fully compatible with North Star Computers, Inc. disk controller at 0E800H and well support the Digital Research, Inc. CP/M system up to a size of 58K. Using the Micro Complex Disk Contoller with address selection option, you can move the controller up to 0EC00H and run a 59K system with no modification of the Lifeboat Associates` 2.21A version of CP/M.

This example is probably the most desirable application of the DPM to most users. However, there are many more applications available to the systems programmer and those desiring to have more control, accessability, and insight to the machine level of software intricacies of the SOL computer. To sevice this requirement, the DPM provides the primary function of supporting a 2716 PROM in socket "B" and in socket "A" either a 2716 or a standard SOLOS PROM. Jumper options are provided to support which ever of the three you choose for socket "B". In addition, options are provided to have the address of both PROMs to be at 0C000H or 0F000H, or socket "A" at 0C000H and socket "B" switches to address 0F000H. These options provide considerable versatility in your selection of mode of operation depending upon your desired application.

The standard configuration for the DPM as shipped, includes two installed 2716 PROMs. These are programmed to provide all the significant SOLOS functions except the cassette routines. There are instead a total of 20 new commands, as documented in Appendix A. These commands occupy nearly half the PROM space and can be replaced with user modified code if so desired on a special order basis. Minor modifications cost about $25.00, more extensive modifications are priced in accordance with the time required to program the requested modifications. If the user provides the code modifications, then the customization will cost between $25.00 and $50.00 for the initial PROM and $15.00 for each additional copy (Includes the cost of 2716 PROM).

The standard configuration of the DPM is shipped with an auto-load jump to address 0E800H which is initiated by

CON'T FROM PAGE 26 - DUAL PERSONALITY MODULE

pressing the "LOAD" key on the SOL. This will auto-load North Star Computer, Inc. compatible disk controllers. At the same price, you may request the DPM version which will auto-boot the Processor Technology, Corp. HELIOS II with either the standard "BO" command or by pressing the "LOAD" key on the SOL. If you are running both a North Star Computer, Inc. type disk controller and a Helios II disk system in your computer, then you might like to have the third version of the DPM which boots the Helios via the "BO" command and boots the North Star by pressing the "LOAD" key.

Instructions for the simple modification required to add two jumpers to the top of your SOL`s circuit board are provided by Appendix B.

Instructions for a two cut modification of your existing SOLOS PM board to allow it to function normally after you add the two jumpers (as per Appendix B) are provided in Appendix C. This will allow you to use your old PM card to read cassette tapes, in the event that you might encounter that requirement.

It should be noted that it is necessary to modify your software programs, which interface with SOLOS, to reference the corresponding functional vectors in the 0F000H support PROM software. The normal entry points relative to the Monitor origin address have been maintained in order to make the requried modifications as easy as possible, ie. a jump to the screen driver would have to be modified to jump to 0F054H vice 0C054H as in the normal SOLOS PROM. These vector are only maintained for the relative low address entries. If your programs require a more exact correspondence to SOLOS, then I refer you to the article on page #8 of the November/December issue of the "PROTEUS" newletter as published by PROTEUS, 1690 Woodside Road, Suite 219, Redwood City, CA. 94061,USA.

Modifications of programs to run with the Monitor at 0F000H will normally be rather minor. However, some projects such as modifing the Helios PTDOS to be fully compatible at 0F000H will require significant work. I would be interested in participating in a project to achieve that goal. It would be worth the effort to have the extended contiguous memory for the various large programs under PTDOS, ie. FORTRAN, WORD WIZARD, and ALS-8 assemblies.

It might make the operation of a lot of your software more flexible if you put your customized Input/Out device code into the PROMs on the DPM. Then you could move the software to new ORG addresses without having to make additional USER I/O modifications. This would be very convient when building different sized CP/M systems for special applications in your machine. Also, use of the new North Ster Computer`s System Disk Release 5.2 would be enhanced, in that, you could use the Mover program to reconfigure your system and not have to make modifications to the USER I/O area for each configuration.

Ed Note: See our article on making SOL's address space toggle in and out, page 1 "DON'T MOVE SOL TO FOOO, MAKE SOL VANISH"

TERM PROGRAM REPLACES SOLOS TERM COMMAND
BY Fr. Thomas McGahee

TERM program is meant to replace the SOLOS TERM command. It allows the user to direct stuff to the printer, assuming the printer is on the parallel port.

The program gets its options from the keyboard. The options include:
1) **LOAD**: this allows toggling the video echo of keyboard input on and off. It comes up OFF initially.

2) **CURSOR-RIGHT**: this enables the printer.

3) **CURSOR-LEFT**: this disables the printer.

4) **CURSOR-UP**: this will allow the printer to print BOTH what is received via the modem AND what is sent via the keyboard (assuming the printer is enabled).

5) **CURSOR-DOWN**: this will allow the printer to print ONLY what is received via the modem. Characters from keyboard will only be printed if the modem system is echoing them.

The initial state is the same as if the LOAD key was toggled OFF, and the CURSOR-UP and CURSOR-RIGHT keys were active.

The reason for supplying all these options is to allow the user to see what HE wants on the screen, and to direct what HE wants to the printer.

A note to software freaks: I now have my latest version of the MODIFIER ready for shipment. The MODIFIER 3 will run on ANY 8Ø8Ø/8Ø85/Z8Ø computer. It will accomodate ANY screen format, including 16x64 and 24x8Ø. It requires memory-mapped video, but the user may specify the address. The standard version comes ready to run on a SOL-2Ø. It is self-relocateable, and comes with source code so it can be modified by the user if desired. Specify whether you want it on SOLOS/CUTER tape for running on a SOL, or on SOLOS/CUTER TAPE in TAPEDISK format for loading to a CP/M system (tape loader program included on tape), or on standard 8" disk for CP/M single density. Sorry, but I cannot provide Northstar, Micropolis, Helios etc., but please note that the TAPEDISK program allows ANY CP/M disk system of ANY density to load in the tape. All versions cost the same amount: $3Ø. I also still have my special Northstar printer drivers available at $1Ø each. (see ad in Volume 3 #4). Please include a description of your system when ordering any software, including what port you are connected to. That way I can give you a program that will run on YOUR machine!

Also, as many PROTEUS persons have discovered, I am usually available for custom software and hardware work. My rates are reasonable, and I generally get things done fast. Give me a call if you need help. I charge nothing for help over the phone... you just pay for the call. Oh yes, If you write requesting some information, PLEASE include a self addressed and stamped envelope. Glad to be of help!

Sincerely yours,
Fr. Thomas McGahee
Fr. Thomas McGahee S.D.B.
(2Ø1) 595-88ØØ (Electronics Dept.)

PS I am also including some pages regarding the relocation of SOLOS ROM/RAM to FØØØH for publication in PROTEUS.

CON'T FROM PAGE 27: - TERMINAL COMMAND

```
                  ;       *****                  *****
                  ;
                  ;       THIS ROUTINE GETS CHARACTERS FROM THE SYSTEM KEYBOARD
                  ; AND OUTPUTS THEM TO THE SELECTED OUTPUT PORT.  IT IS
                  ; INTENDED TO CONFIGURE THE SOL AS A STANDARD VIDEO
                  ; TERMINAL.  COMMAND KEYS ARE NOT OUTPUT TO THE OUTPUT
                  ; PORT BUT ARE INTERPRETED AS DIRECT SOL COMMANDS.
                  ; THE MODE COMMAND, RECEIVED BY THE KEYBOARD, PUTS THE SOL
                  ; IN THE COMMAND MODE.
                  ;
                  ; *** ENHANCEMENTS ADDED BY FR. THOMAS MCGAHEE ***
                  ; ENHANCEMENTS INCLUDE:
                  ; CURSOR-RIGHT ENABLES CENTRONICS PRINTER.
                  ; CURSOR-LEFT DISABLES CENTRONICS PRINTER.
                  ; CURSOR-UP ALLOWS ECHO OF BOTH KEYBOARD AND SERIAL PORT.
                  ; CURSOR-DOWN ALLOWS ECHO OF SERIAL PORT ONLY.
                  ; LOAD KEY TOGGLES VIDEO ECHO OF KEYBOARD.
                  ;
                  ; *** FR. THOMAS MCGAHEE
                  ; *** DON BOSCO TECH
                  ; *** 2Ø2 UNION AVE.
                  ; *** PATERSON, NJ Ø75Ø2
                  ; *** 2Ø1-595-88ØØ
                  ;
                  ; *** JULY 23, 1981
                  ;
                  ;
                  ;
CØØØ =            SOLOS    EQU     ØCØØØH  ;BASE OF SOLOS ROM.
CØ1F =            SINP     EQU     SOLOS+1FH
CØ19 =            SOUT     EQU     SOLOS+19H
CØ2E =            KSTAT    EQU     SOLOS+2EH
C8Ø6 =            IPORT    EQU     SOLOS+8Ø6H
C8Ø7 =            OPORT    EQU     SOLOS+8Ø7H
C1CØ =            COMN1    EQU     SOLOS+1CØH
CØ54 =            VDMOT    EQU     SOLOS+54H
C2E6 =            CENTRONICS EQU   SOLOS+2E6H
C8ØC =            ESCFL    EQU     SOLOS+8ØCH
C812 =            KBDFLAG EQU      SOLOS+812H
C813 =            CENTFLAG EQU     SOLOS+813H
C814 =            BOTHFLAG EQU     SOLOS+814H
C815 =            TOGGLE   EQU     SOLOS+815H
                  ;
ØØ8Ø =            MODE     EQU     8ØH      ;MODE=TERMINATE
ØØ8C =            LOAD     EQU     8CH      ;LOAD=TOGGLE VIDEO ECHO
ØØ97 =            CURSUP   EQU     97H      ;CURSOR-UP=VIDEO+CENTRONICS
ØØ9A =            CURSDWN EQU      9AH      ;CURSOR-DOWN=VIDEO ONLY
ØØ81 =            CURSL    EQU     81H      ;CURSOR-LEFT=CENTRONICS OFF
ØØ93 =            CURSR    EQU     93H      ;CURSOR-RIGHT=CENTRONICS ON
ØØ1B =            ESC      EQU     1BH
ØØØD =            CR       EQU     ØDH
ØØØA =            LF       EQU     ØAH
                  ;
                  ;
                  ;
ØØØØ                       ORG     ØØØØH   ;RUNS AT ZERO.
                  ;
                  ;
                  TERM:
ØØØØ 3EØ1                  MVI     A,1      ;SERIAL PORT IS I/O CHANNEL
ØØØ2 32Ø6C8                STA     IPORT
ØØØ5 32Ø7C8                STA     OPORT
ØØØ8 3214C8                STA     BOTHFLAG ;START WITH BOTH ENABLED
ØØØB 3213C8                STA     CENTFLAG ;START WITH CENTRONICS ON.
ØØØE 3EØØ                  MVI A,Ø
ØØ1Ø 3215C8                STA     TOGGLE   ;START WITH VIDEO ECHO DISABLED.
                  ;
ØØ13 CD2ECØ       TERM1:   CALL    KSTAT    ;CHECK KEYBOARD.
ØØ16 CA77ØØ                JZ      TIN      ;IF NOTHING, CHECK SERIAL PORT.
ØØ19 47                    MOV     B,A      ;SAVE IT IN B
```

```
ØØ1A 3EØ1                  MVI      A,1
ØØ1C 3212C8                STA      KBDFLAG ;SAY IT'S FROM KEYBOARD.
ØØ1F 78                    MOV      A,B
ØØ2Ø FE8Ø                  CPI      MODE     ;IS IT MODE?
ØØ22 CACØC1                JZ       COMN1    ;YES...RESET AND QUIT TERM
ØØ25 DA71ØØ                JC       TOUT     ;NON-CURSOR KEY...SEND TO TERM PORT
ØØ28 FE8C                  CPI      LOAD     ;IS IT LOAD KEY ?
ØØ2A C237ØØ                JNZ      CURL
ØØ2D 3A15C8                LDA      TOGGLE   ;GET VIDEO TOGGLE
ØØ3Ø 3C                    INR      A        ;TOGGLE IT.
ØØ31 3215C8                STA      TOGGLE
ØØ34 C313ØØ                JMP      TERM1
ØØ37 FE81         CURL:    CPI      CURSL    ;IS IT CURSOR-LEFT?
ØØ39 C244ØØ                JNZ      CRR
ØØ3C 3EØØ                  MVI      A,Ø
ØØ3E 3213C8                STA      CENTFLAG          ;FLAG CENTRONICS OFF.
ØØ41 C313ØØ                JMP      TERM1
ØØ44 FE93         CRR:     CPI      CURSR    ;IS IT CURSOR-RIGHT?
ØØ46 C251ØØ                JNZ      CUP
ØØ49 3EØ1                  MVI      A,1
ØØ4B 3213C8                STA      CENTFLAG          ;FLAG CENTRONICS ON.
ØØ4E C313ØØ                JMP      TERM1
ØØ51 FE97         CUP:     CPI      CURSUP   ;IS IT CURSOR-UP?
ØØ53 C25EØØ                JNZ      CDWN
ØØ56 3EØ1                  MVI      A,1
ØØ58 3214C8                STA      BOTHFLAG          ;FLAG BOTH
ØØ5B C313ØØ                JMP      TERM1
ØØ5E FE9A         CDWN:    CPI      CURSDWN ;IS IT DOWN-CURSOR?
ØØ6Ø C26BØØ                JNZ      REG
ØØ63 3EØØ                  MVI      A,Ø
ØØ65 3214C8                STA      BOTHFLAG          ;FLAG ONLY SERIAL
ØØ68 C313ØØ                JMP      TERM1
                  REG:
ØØ6B CD54CØ                CALL     VDMOT    ;PROCESS IT
ØØ6E C377ØØ                JMP      TIN

                  ;
ØØ71 CD19CØ       TOUT:    CALL     SOUT     ;OUTPUT IT TO THE SERIAL PORT
ØØ74 CDCCØØ                CALL     VID      ;ECHO ON VIDEO?
ØØ77 CD1FCØ       TIN:     CALL     SINP     ;GET INPUT STATUS
ØØ7A CA13ØØ                JZ       TERM1    ;IF NOTHING, THEN CHECK KBD.
ØØ7D E67F                  ANI      7FH      ;NO HIGH BITS FROM HERE
ØØ7F CA13ØØ                JZ       TERM1    ;A NULL IS IGNORED
ØØ82 47                    MOV      B,A      ;IT'S OUTPUT FROM 'B'
ØØ83 3EØØ                  MVI      A,Ø
ØØ85 3212C8                STA      KBDFLAG ;SAY IT'S NOT FROM KBD.
ØØ88 78                    MOV      A,B
ØØ89 FE1B                  CPI      1BH      ;IS IT A CONTROL CHAR TO BE IGNORED
ØØ8B D2ABØØ                JNC      TERM2    ;NO...TO VDM AS IS THEN
ØØ8E FEØD                  CPI      CR       ;CR OR LF ARE SPECIAL CASES THOUGH
ØØ9Ø CAABØØ                JZ       TERM2    ;AND MUST BE PASSED STD MODE TO VDM
ØØ93 FEØA                  CPI      LF
ØØ95 CAABØØ                JZ       TERM2
ØØ98 3AØCC8                LDA      ESCFL    ;A CTRL CHAR...ARE WE W/IN ESC SEQUENCE?
ØØ9B B7                    ORA      A        ;IF YES, THEN OUTPUT CTRL CHAR DIRECTLY TO
ØØ9C C2ABØØ                JNZ      TERM2    ;WE SURE ARE, LET VDM DRIVER HANDLE IT
ØØ9F C5                    PUSH     B        ;SAVE THE CHARACTER
ØØAØ Ø61B                  MVI      B,ESC    ;CTRL CHAR TO VDM VIA ESC SEQUENCE
ØØA2 CD54CØ                CALL     VDMOT
ØØA5 Ø6Ø7                  MVI      B,7      ;SAY TO PUT OUT NEXT CHAR AS IS
ØØA7 CD54CØ                CALL      VDMOT   ;ALMOST READY
ØØAA C1                    POP      B        ;RESTORE CHAR
ØØAB =            TERM2:   EQU      $        ;ALL READY TO OUTPUT THE CHAR
ØØAB CD54CØ                CALL     VDMOT    ;PUT IT ON THE SCREEN
ØØAE CDB4ØØ                CALL     ECHO
ØØB1 C313ØØ                JMP      TERM1
                  ECHO:
ØØB4 3A13C8                LDA      CENTFLAG
ØØB7 FEØØ                  CPI      Ø
ØØB9 C8                    RZ                ;IF NO-PRINT, DON'T PRINT.
ØØBA 3A14C8                LDA      BOTHFLAG
```

CON'T FROM PAGE 28 - TERMINAL COMMAND

```
ØØBD FEØ1                  CPI     1
ØØBF CAC8ØØ                JZ      TERM3   ;IF BOTH ARE OK, PRINT IT.
ØØC2 3A12C8                LDA     KBDFLAG
ØØC5 FEØ1                  CPI     1       ;IF BOTH ARE NOT OK, THEN
ØØC7 C8                    RZ              ;IF KEYBOARD, DON'T PRINT.
                 TERM3:
ØØC8 CDE6C2                CALL    CENTRONICS       ;PRINT USING CENTRONICS.
ØØCB C9                    RET
                 ;
                 ;
                 ;
ØØCC 3A15C8      VID:      LDA     TOGGLE
ØØCF E6Ø1                  ANI     1       ;VIDEO ECHO ON "ODD" TOGGLES ONLY.
ØØD1 C8                    RZ              ;(ON EVEN TOGGLES DO NOTHING).
ØØD2 CD54CØ                CALL    VDMOT   ;PUT IT ON SCREEN IF ODD TOGGLE.
ØØD5 CDB4ØØ                CALL    ECHO    ;MAYBE PRINT IT TOO.
ØØD8 C9                    RET
```

RELOCATION OF SOLOS TO FØØØ

By
Father Thomas McGahee
Don Bosco Technical High School
2Ø2 Union Ave.
Paterson, NJ Ø75Ø2

The purpose of this paper is to explain how to convert a SOL-2Ø so that the SOLOS operating system ROM resides at FØØØH instead of its original CØØØH base address.

The primary reason for relocating SOLOS to FØØØH is to allow a larger area of contiguous RAM to be available. Please note that the following relocation will also automatically relocate the VIDEO DISPLAY MEMORY to FCØØH, the SYSTEM GLOBAL RAM to F8ØØH, and the VIDEO DISPLAY MEMORY to FCØØH.

HARDWARE CHANGES

There are few hardware changes that need to be made. None of these require any changes to the SOL motherboard. Instead, the changes are made directly at the Integrated Circuits involved. The first change to be made is to U22. Remove it from its socket, and carefully add a jumper between pins 5 & 2 of this IC. (An alternative is to jumper pins 9 & 12... either approach works). I added the jumper using a small length of wire-wrap wire. To avoid getting solder on the pins, I turned the IC upside down and carefully applied a SMALL amount of solder to the shoulder of the IC pin. I then tinned the ends of the wire-wrap wire and gently soldered it to the shoulders. I re-installed the IC once the jumper was added.

The second change involves removing U24. Replace it with a 74SØ4 that has been prepared by adding a 1.5K 1/4 watt resistor between pins 1 & 14. (This provides active pullup for the PHANTOM line). The 74SØ4 is used because the original IC may not be able to handle the increased load that resulted when we jumpered pins 5 & 2 of U22. I bent the leads of the resistor so that they just touched the shoulders of the IC at the proper pins, then cut the resistor leads off so that they did not extend beyond the broad part of the shoulder. I then tinned the shoulders (with IC upside-down to prevent solder flow onto the pins themselves), tinned the resistor leads, then soldered the resistor to the shoulders of pins 1 & 14. I then re-installed U24.

PERSONALITY MODULE CHANGES

If you have a personality module that is currently using 27Ø8s, then simply install the new 27Ø8s in their place. The 27Ø8s are labelled U1 and U2. U1 is the IC closest to the side of the case.

If you have a 27Ø8 module that has a single 9216 or 8316 ROM, then you will have to make one minor modification to the module to allow it to be used with a 2716 EPROM.

On the edge of the personality module nearest U1 you will find a series of feed-through holes. Locate the ones labelled "19" and "+12". There is a run connecting these two on the BOTTOM of the board. CUT THIS RUN. This disconnects the +12 volts from pin 19 of U1. Now solder a jumper between the feedthroughs labelled "19" and "CØ". This connects the proper chip select signal to pin 19 of U1.

Remove the original 27Ø8s or 9216, and replace with the new 27Ø8s or 2716 EPROM. This completes all necessary changes. Insert the personality board back into its connector and power up your SOL. If you have performed all the changes properly, your system should come up with the ">" prompt symbol.

HINTS

Once the system is relocated to FØØØH, software written for use at CØØØH will not run properly. For much of this software a simple solution is to place into memory a JUMP TABLE at CØØØH that is identical to that at FØØØH. This can be easily done by typing "SAVE JMP FØØØ FØ2D CØØØ". This tape copy includes the jump table and the first part of the code for AINP. When this tape is loaded by typing "GET JMP", it will cause all calls to the table at CØØØH to access the routines at FØØØH.

For example, let's say you have a CP/M operating system currently configured for 46K that expects to find SOLOS at CØØØH, and you have just relocated SOLOS to FØØØH. Load the tape copy of the FØØØH jump table down to CØØØH by typing "GET JMP". Now you can boot up your disk with no problems. Once you have the disk system running, you can edit your BIOS so that all references to addresses in the range CØØØH-CCØØH now reference the FØØØH-FCØØH system ROM/RAM area. At the same time you can change the system size to a larger value, such as 6ØK. Assemble the new BIOS and integrate it into a new operating system (follow the instructions in the CP/M guide). Once you have done this and placed a copy of the new system on disk using SYSGEN, then you can boot up the new system anytime you want.

A few words of caution. Some disk systems use an extended BIOS, in which case the ACTUAL memory size needed EXCEEDS

CON'T FROM PAGE 29: - RELOCATION OF SOLOS TO F000

the "NOMINAL" memory size by one or two K. Thus some users will have to declare systems of only 59K or 58K instead of 60K. Also, some disk systems are MEMORY MAPPED. This creates a problem if the memory so mapped falls within the space occupied by the relocated SOLOS/RAM/VIDEO combination. This is all memory from F000H-FFFFH. To use such a system with the relocated SOLOS will require re-mapping of the disk controller and appropriate changes to the controller software. Controllers that are I/O mapped will work without any modifications at all, since they occupy no address space.

Any program that you have that uses direct access to the VIDEO RAM will not run under the relocated system unless changes are made to the program in question. Keep in mind that the VIDEO RAM is now from FC00H-FFFFH. To run programs such as the ELECTRIC PENCIL or DDS will require some patching.

You will notice that it is possible to do a little rework on a 2708 Personality Module such that U1 could contain the original 9216 PROM, and U2 could contain the 2716 EPROM. A double-pole double-throw switch could be wired in to select either U1 or U2 as desired. (Do this by switching in the chip select pin). The same switch could be used to activate or deactivate the jumper on U22. In this way the switch could be used to select either the original SOLOS at C000H, or the SOLOS at F000H. This would allow all current software to be run unchanged PROVIDED SOME METHOD IS USED TO DEACTIVATE PLUG-IN MEMORY FROM C000H-CFFFH WHENEVER THE C000H SOLOS IS ACTIVE. Perhaps an unused buss line could be used to phantom out the memory by deactivating the memory's chip select logic.

Included with this documentation is a complete source listing of the relocated version of SOLOS. This version retains all the major features of the original SOLOS, but with a few enhancements. The TERMINAL command has been replaced with a BOOT command for use by HELIOS owners. Underline and backspace now work properly. Under CP/M, lines will no longer disappear from the screen when double carriage-returns are received.

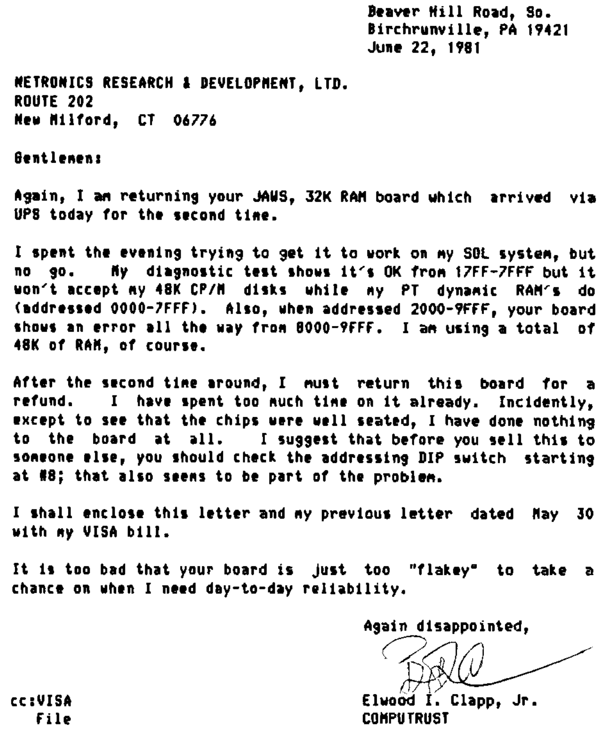
It is my sincere hope that this relocated version of SOLOS will be useful to you as you begin the task of upgrading your SOL-20 to a larger memory capacity.

RELOCATED SOLOS EPROMS AND DOCUMENTATION $40.
REQUIRES 2708 PERSONALITY MODULE.
SPECIFY 2708s OR 2716 EPROM.

ORDER FROM: FR. THOMAS MCGAHEE
DON BOSCO TECH
202 UNION AVE.
PATERSON, NJ 07502 (201) 595-8800

A BAD REPORT
BY E.I. Clapp, Jr.

"Stan- The attached represents a frustating experience - not to mention about $10.00 in long distance calls and postage. Perhaps others should be cautious in purchasing this item.
P.S. - this was an assembled and tested board.
Elwood I. Clapp, Jr."



Beaver Hill Road, So.
Birchrunville, PA 19421
June 22, 1981

METRONICS RESEARCH & DEVELOPMENT, LTD.
ROUTE 202
New Milford, CT 06776

Gentlemen:

Again, I am returning your JAWS, 32K RAM board which arrived via UPS today for the second time.

I spent the evening trying to get it to work on my SOL system, but no go. My diagnostic test shows it's OK from 17FF-7FFF but it won't accept my 48K CP/M disks while my PT dynamic RAM's do (addressed 0000-7FFF). Also, when addressed 2000-9FFF, your board shows an error all the way from 8000-9FFF. I am using a total of 48K of RAM, of course.

After the second time around, I must return this board for a refund. I have spent too much time on it already. Incidently, except to see that the chips were well seated, I have done nothing to the board at all. I suggest that before you sell this to someone else, you should check the addressing DIP switch starting at #8; that also seems to be part of the problem.

I shall enclose this letter and my previous letter dated May 30 with my VISA bill.

It is too bad that your board is just too "flakey" to take a chance on when I need day-to-day reliability.

Again disappointed,

cc:VISA
File

Elwood I. Clapp, Jr.
COMPUTRUST

HELP NEEDED ON UCSD PASCAL ON S-100
BY Kerry Montgomery

5-23-81
"Hello -

As a 2 year subscriber to Proteus, I have a request for help. I'm trying to bring up UCSD PASCAL on an S-100 system that has Processor Technology's Subsystem-B boards. The problem is in interfacing USCD PASCAL's setup and X-Y cursor addressing schemes with the VDM-1. All seems well except the UCSD editor is not ueable. Can you or any members of Proteus offer any help?
Thanks,
Kerry Montgomery
13420 S.W. Castlewood
Beaverton, OR 97005"

MORE LETTERS TO THE EDITOR:

Word Works
6 Bow Street, Cambridge, MA 02138, 617/491-3999

April 23, 1981

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

Dear Mr. Sokolow:

I have a SOL-20 with a 16KRA board that has a bad delay line on the memory board itself. The legend printed on the delay line is "Data Delay DDU-4-7672". I have been unable to locate this part, and I would appreciate it if you could recommend a possible source to me. A scribbled note on this letter returned in the enclosed return envelope would certainly be sufficient. Thanks muchly.

Sincerely,

Edwin Meyer

Edwin W. Meyer

Dear Stan,

I have been looking for a graphics board for my SOL, but it is very difficult to tell exactly how they are from magazine ads. If some other SOL users have graphics I would appreciate some help in choosing the right graphics for my system.

Thank you,

Mac Watts
710 Norman Cir.
Opelika, AL 36801

6-MAY-81

Dear Stan;

I'm one of the isolationists with my SOLOS at FOOOH and my software sitting in object code looking vainly for COOOH. To anyone with the ELECTRIC PENCIL ver.SS changing ODF5H from CC to FC makes it feel right at home in my 60K CP/M system. Now if someone can find a way to make it pass control characters to my EPSON MX-80 printer the "PENCIL" will live on! Thanks for a great (and needed) publication.

Bob Johnson
565 Mohawk Dr. A-5
Boulder,Colo.80303

May 7,1981

Stan Sokolow
PROTEUS
1690 Woodside Rd., Suite 219
Redwood City, CA 94061

Dear Stan,

It has been awhile since I wrote to PROTEUS but now it's so much easier - I took the big printer plunge. I just recently purchased from MICROPERIPHERALS, INC. their MODEL 88G. It is a slick little dot-matrix printer. This letter was printed by it, in its correspondence font (11x7 matrix). It has alot of neat features - many of which are software controllable. I also learned, with the help of previous letters in PROTEUS, about printer drivers. At first I had it interfaced through my serial port on my SOL, but I use a modem and did not want to give up that aspect of computing. My parallel port was free and SOLOS supports the 'HANDSHAKING' that the 88G needed to operate properly, so I used the parallel port and it works great. Other members just might be ready to purchase an excellent-quality printer without going bankrupt. The MODEL 88G lists for $749.00, but substantial discounts are available from many of the mail order houses listed in many of the magazines. By the way, the 88G has very good, dot-addressable graphics for those who might need it. It's bidirectional too.

On another note - I currently use a cassette version of the ELECTRIC PENCIL I for letters such as this one, and various other memos. The cassette SAVE and LOAD routines are extremely slow. Has anyone out there in SOL / NORTHSTAR land been successful in patching the cassette routines in the ELECTRIC PENCIL to the N* disk routines. I don't think Michael Shrayer is marketing the N* version anymore, does anyone know?

I also have been using the ADS NOISEMAKER board for about six months now - I have written a program in PT's ECBASIC that will allow changes to all of the registers - dynamically to. When I get it in publishable form I will send it in. It works very effectively.

Enough of my rambling - keep up the excellent work that you so faithfully and diligently put forth to keep this the BEST newsletter in our industry. Many thanks also to all who contribute letters on a regular basis. I look forward every other month with much anticipation to receive PROTEUS NEWS.

Thanks Again.

Rick

Rick Downs
P.O. BOX 440357
AURORA, CO 80044
(303 750-1838

FLOPPY DISK SERVICES, INC.
Siemens Computer Disk Drives
C.N. 5212 PRINCETON, NEW JERSEY 08540

TELEPHONE 609/771-0374

Here is the information you requested. Thank you for your interest in our disk drives. All drives are brand new and warranted for 90 days from your date of shipment. Please feel free to call and ask any questions about interface or other technical info.

You may also be pleased to know that we may soon have hard disk systems available ready to run in your S-100 systems.

Remember that our 5 1/4" drives are the 'Flippy' models. Most other stores charge $20 to $35 more for this option that is standard on our 5 1/4" drives!

FDD-100-8d2 (8" single sided, single or double density)......	$360.00
FDD-100-5b4 (5 1/4" single sided 'FLIPPY' DD or SD)..........	$250.00
Maintenance manuals 8" or 5 1/4".............................	$13.00
Power connector set (1-AC,1-DC 8" drives)...................	$3.50
Power connector 5 1/4"......................................	$1.00
Edge card connectors 50 or 34 pin...........................	$10.00
Cables made to your spec....................................	CALL
** 2-8" drive package...........................with cables	$995.00
** 1-8" drive package.(1 drive in dual cabinet)...with cables	$665.00
1-51/4" flippy case,power supply tested..................	$295.00
2-51/4" flippies dual case, power supply tested..........	$595.00

* Shipping NOT INCLUDED ** 8" packages require assembly
All prices and specs subject to change without notice
NJ residents add 5% sales tax

PAYMENT POLICY

We accept Mastercard, Visa, personal checks, MO, COD with PRIOR PERMISSION ONLY, (CASH will be required). We reserve the right to wait 10 working days for personal checks to clear your bank before we ship. Shipping charges MUST be included or your order will be delayed. All charges are standard UPS rates plus insurance.

(317)448-1616 • TWX 810-342-1881
Micro Data Base Systems, inc.
P.O.BOX 248 LAFAYETTE, IN 47902

NEWS RELEASE: NEW LITERATURE

Micro Data Base Systems, Inc. (Box 248, Lafayette, IN 47902) now provides a free decision pamphlet upon request. This pamphlet identifies and describes the important features to consider when deciding upon a software tool to use for application system development. The features discussed include the richness of logical data structuring supported (are all kinds of data relationships easily represented, can the semantics of these relationships be clearly indicated, etc.), control over data redundancy and data inconsistency, availability of a high level query language, availability of programming language interfaces, mechanisms for data base recovery and restructuring, and portability of the software tool across machines and operating systems (allowing a standardized approach to application development).

Application development tools range from flat file systems to genuine data base management systems. There are several varieties of data base management systems including hierarchical, CODASYL networks, and the recent innovation of extended networks as provided by MDBS. Regardless of which flat file or data base management systems an application developer is considering, they should be examined in terms of the features described in the decision pamphlet. A convenient feature check list is provided in the pamphlet.

####

CONTACT: Mr. John Morris
Sales Manager

SUNBELT COMPUTER EXPO

SUNBELT COMPUTER EXPO
SEPTEMBER 19, 20, 21, 1981, PHOENIX CIVIC PLAZA, PHOENIX, AZ

The purpose of the SUNBELT COMPUTER EXPO '81 is to provide an annual forum for computer professionals in all fields to share their knowledge, experience, and research developments. The focus of the 1981 SUNBELT COMPUTER EXPO is computer application in buiness, industry, word processing, education, and home use.

If you are interested in presenting a seminar contact:
Kerry S. Judd, President
Judco Enterprises
Post Office Box 963
Scottsdale, AZ 85252 (692)991-8622.

MICRO DYNAMICS

4444 SCOTTS VALLEY DRIVE, SCOTTS VALLEY, CALIFORNIA 95066 • (408) 438-5454

June 11, 1981

MICRO DYNAMICS has used floppies available again. These discs have been used as files for diagnostic studies in medical institutions. Most have had only limited use and have been handled carefully. The discs are 8" single density with a mix of single sided and reversible configurations.

Discs are priced at $ .95 each and must be ordered in cases of 50 discs. UPS shipping charges must be added to the order and California residents must add sales tax.

We are also a full line micro-computer accessory dealer and can supply all of your magnetic media and media storage needs, for example plastic file boxes and pages for floppy discs or Scotch® Head Cleaning Kits.

Remember, these discs are in limited supply and will go quickly. Orders will be filled on a first come, first serve basis. Payment can be made by money order, cashiers check, VISA or MasterCard. Place your order early to take advantage of this incredible price.

Sincerely,

Les Inanchy
General Manager

CONSUMER COMPLAINTS
SD Sales Expandoram I
Micro Complex Dual Personality Module

"I have an Expandoram I from SD Sales and I'm here to tell you that it's probably the worst investment I ever made. It has 64K of memory and can't even remember who it is, much less anything else. I think, in view of the fact that so many people have had the same experience, SD owes us all a lot of explaining.

"On Micro Complex's Dual Personality Module, be advised that no tape commands are present, and those easy changes to your software to move from C000 to F000 aren't so simple if you're not into machine language and a professional programmer."

Bruce G. Diller, 18651 E. Gallarno Dr., Covina, CA 91722.

Editor's Note: Micro Complex has replaced several Solos commands with other commands they felt are more useful in a disk system. This custom version of Solos can be ordered in several configurations, so I believe that Micro Complex will program the EPROM with standard Solos for you as a special order. See article in this issue on "vanishing" Solos for another solution to the problem.

Owners of SD Sales Expandoram I with complaints are requested to write an explanation of the problem to Proteus. We have had items in the past on a slight modification needed to make Expandoram compatible with Sol.

Stretching Microcomputer Power With ANS MUMPS

This 90-page special issue of the MUMPS Users' Group quarterly publication describes in detail the more than fifteen MUMPS implementations available for microcomputers, including 8080, 8085, and Z80-based micros, the TRS80 Model II, the Motorola 6809, DEC's LSI-11/2, 11/3, and 11/23, the Data General MicroNOVA, and the IBM Series 1. It includes introductory information about the MUMPS programming language and about microcomputers. Detailed technical descriptions are given of MUMPS implementation techniques and special characteristics of microcomputer MUMPS implementations. MUMPS microcomputer applications currently in use are reviewed.

This single special issue sells for $7.25. Membership in the MUMPS Users' Group for 1980-81 costs $35 and includes four issues of the MUG Quarterly: The Micros Are Coming, Stretching Microcomputer Power with ANS MUMPS, the special Vendor Issue, and the Business Applications Issue. Members are also entitled to discounts on other MUG publications.

Partial CONTENTS

The Micro Breakthrough
Pascal, Basic and MUMPS
The Anatomy of a Microcomputer
Special Characteristics of Microcomputer MUMPS Implementations
Microprocessor MUMPS Implementation Techniques
MUMPS Implementations for Microcomputers
Evaluating Types of Applications Which Will Run on Microcomputer MUMPS
MUMPS Applications on Microcomputers
Automated Instruments and MUMPS -- The Microcomputer as Interface
MUMPS on the Micro: Thoughts and Experiences
Proposal for a MicroMUMPS Users' Group

- - - - - - - - - - - - - - - - - - ORDER FORM -

Name____________________________

Address_________________________

____

Telephone_______________________

Make checks payable to:

MUMPS Users' Group

Payment must be enclosed with order.

Mail to: MUMPS Users' Group
1001 Connecticut Avenue, NW
Suite 800
Washington, DC 20036
Tel. (202) 659-5933

Enclosed is:

□ $____for___copies of Stretching Microcomputer Power with ANS MUMPS

□ $35 for MUG membership, 1980-81

PROTEUS CASSETTE SOFTWARE LIBRARY CATALOG

Contents of Proteus Cassette #1 -- Extended Cassette BASIC

```
PLOTS C 3C20 00D1 Plots sine & cosine.
XYPTO C 3C20 0167 An aid for decoding cryptograms.
MTCHS C 3C20 0357 The game of Nim; you against computer.
LUNAR C 3C20 097A Lunar lander simulation; you are pilot.
HXDEC C 3C20 022F Converts hexidecimal numbers to decimal.
MATH  C 3C20 06CF Arithmetic practice for children.
FINAN C 3C20 15E9 A home financial accounting program.
VDMFB C 3C20 1BA5 Video football.
PNUTS C 3C20 0F95 Draws Peanuts cartoon characters.
SLOTS C 3C20 0461 Simulates a slot machine.
SORTS C 3C20 0D61 Compares the speed of 3 sorting methods.
ALFA  C 3C20 0486 Alphabetizes words.
DAYS  C 3C20 03C2 Computes number of days between any 2 dates.
NAMES C 3C20 081E Alphabetizes names.
SLOTM C 3C20 0A11 Another slot machine.
END   C 3C20 0026 End-of-files
```

Contents of Proteus Cassette #2 -- Extended Cassette BASIC

```
KENO  C 3C20 1404 Keno gambling game.
KING  C 3C20 1840 You are king; manage your kingdom wisely.
BIOCH C 3C20 077C Your own biorhythm chart.
MONOP C 3C20 35BB A "Monopoly"-like game.
BLKJK C 3C20 141F Blackjack card game, with graphic display.
STORY C 3C20 0B11 Create a children's story.
SILLY C 3C20 0BF2 Create a silly children's story.
WMPUS C 3C20 0E79 Hunt the "wumpus" in his caves.
END   C 3C20 0026 End-of-files
```

Contents of Proteus Cassette #3 -- Extended Cassette BASIC

```
LIM   C 3C20 0D59 Create a silly limerick.
BAGEL C 3C20 082B A skill-guessing game.
GUESS C 3C20 01FF Number guessing.
MGWMP C 3C20 09EF Find the mugwump; practice in coordinates.
MATCH C 3C20 1006 A word game.
TACTO C 3C20 08AB Tic-tac-toe.
DCODE C 3C20 017B Helps decode cryptograms.
XAGON C 3C20 0A69 A letter capture game; figure out the rules.
MMIND C 3C20 05CF The Mastermind logical guessing game.
UCAL  C 3C20 039A Universal calendar; gives day-of-week for any date.
FSCBA C 3C20 0949 Cost-benefit analysis of fuel-saving investment.
LEASE C 3C20 0609 Computes lease payments, costs, etc.
F2C   C 3C20 00A8 Converts Fahrenheit to Celsius.
C2F   C 3C20 009A Converts Celsius to Fahrenheit.
END   C 3C20 0026 End-of-files
```

Note: Tapes 1, 2, and 3 have each program recorded "compiled" on side 1 (two copies each) and in "text" once on side 2. The "text" form can be read into any system using the Solos/Cuter byte-oriented file operations and they will look like they should in source code of BASIC. The "compiled" files will only make sense if read by Processor Technology Extended Cassette BASIC.

Contents of Proteus Cassette #4 -- Software Tech Music Selections

```
CNTRY   08D3 04BE          GIGUE   08D3 099C
COKES   08D3 0436          COTYR   08D3 0976
SCARB   08D3 0804          SONAT   08D3 0E43
AQUAR   08D3 093A          RAIN    08D3 0964
ORNGB   08D3 0486          LIGHT   08D3 038A
STLIF   08D3 05F8          HEART   08D3 0484
SONGS   08D3 0677          LSTRY   08D3 0377
THING   08D3 037A          CLOSE   08D3 0490
GREEN   08D3 0439          TANGO   08D3 08A9
SILVR   08D3 03DD          GREN2   08D3 0439
NOSUN   08D3 038D          LAURA   08D3 06BC
YANKE   08D3 04FB          PURPL   08D3 06F2
BACH    08D3 1325          YAKY    08D3 0857
WEEKD   08D3 087A          WORLD   08D3 03F0
MICHL   08D3 050D          MOZRT O 08D3 0C43
HERE    08D3 0444          WACHT O 08D3 11DF
RINGO   08D3 04E9          GDAN1 O 08D3 0583
PRCEL   08D3 055F          GDAN2 O 08D3 0583
2PI12   08D3 07AD          GDAN3 O 08D3 04F7
                           MAREF O 08D3 0557
```

(continued at right)

Tape 4 has each music selection recorded once on each side. These require the Software Technology/Processor Technology music system (no longer manufactured).

Contents of Proteus Cassette #5 -- BASIC/5 programs and P.A. tiny Basic

Side 1 -- BASIC/5 programs

```
SLOTS  1AD9 128E Slot machine simulation.
GSTAR  1AD9 04BE Guessing game.
BLKJK  1AD9 1784 Blackjack.
CRAPS  1AD9 07BD Craps.
ACDUC  1AD9 07DF Acey-Ducey.
MMIND  1AD9 1945 Mastermind game.
KING   1AD9 1E18 You are King; manage your kingdom.
SQUIZ  1AD9 11D8 Children's literature quiz.
STARS  1AD9 0A15 Shooting stars game.
TRAP   1AD9 0587 A number guessing game.
TAXMN  1AD9 0DCC A number factoring game.
REVRS  1AD9 0880 Number manipulation game.
HURKL  1AD9 0A33 Find the Hurkle in the coordinate plane.
TTTTT  1AD9 0EAE 3-dimensional Tic-Tac-Toe; challenging!
```

Side 2 -- Palo Alto Tiny BASIC programs

```
BLKJK O 0000 2000 Blackjack in Palo Alto Tiny Basic, includes BASIC.
BTREK O 0000 2000 Tiny Star-Trek game in P.A. Tiny Basic, with BASIC.
```

Note: Palo Alto Tiny BASIC has been documented in the old SCCS INTERFACE magazine (with source listing) and probably in Dr. Dobbs Journal, too; but we don't recall the particular issue. You can see the statements it recognizes by doing an ASCII dump of it. It resides in the first 2K. It supports only integers. No "BYE" command--reset to get back to SOLOS/CUTER.

Tape 5 has each program recorded twice.

PROTEUS CASSETTE SOFTWARE LIBRARY CATALOG

PROTEUS CASSETTE C6

PTC ECBASIC programs--side 1 compiled, side 2 text format.

```
MULTP C 3C20 058A  Multiplication practice
MULTP C 3C20 058A
ADD   C 3C20 0587  Addition practice
ADD   C 3C20 0587
SAIL  C 3C20 0582  Navigate your boat to the islands
SAIL  C 3C20 0582
CHOMP C 3C20 09DA  Force the computer to chomp the last bite
CHOMP C 3C20 09DA    (a 2-dimensional variation of Nim)
ROCKT C 3C20 1399  A sophisticated lunar-lander simulation
ROCKT C 3C20 1399
STOCK C 3C20 1230  A stock market simulation game
STOCK C 3C20 1230
CYLON C 3C20 093D  A pursuit game in real-time.
CYLON C 3C20 093D
SCIFI C 3C20 0E82  Writes all-too-familiar science fiction plots
SCIFI C 3C20 0E82
SPEED C 3C20 0D70  Speed reading competition
SPEED C 3C20 0D70
AARON C 3C20 0BE5  A memory game (tricky)
AARON C 3C20 0BE5
TRUCK C 3C20 1700  Keeps track of vehicle operating costs in
TRUCK C 3C20 1700    several categories
FLOW  C 3C20 07CA  Estimates how long a supply of water will last
FLOW  C 3C20 07CA
END   C 3C20 0019  End-of-files.
END   C 3C20 0019
```

Order from:

Lewis Moseley, Jr.
2576 Glendale Court, NE
Conyers, Georgia 30208

Prices;

$18.00 U.S. without contributed program

$8.00 U.S. with acceptable contributed program

PROTEUS CASSETTE C7
Assembly language source and object

```
SCS16   0100 2700 Assembly language system (Self-Contained System)
SCS16   0100 2700   Editor, assembler, disassembler.  Version 1.6
CSEL  D C863 035E Output driver for IBM 2741 Correspondence code
CSEL  D C863 035E   selectric terminal.  EX C863; SET CO C903; SET O=3.
CUPO    0001 055A Cassette Utility Package.  Introductory documentation
CUPO    0001 055A   is in CUPO as ALS-8 file.
CUP1    0001 1C97 Cassette Utility Package.  Documentation of CUP as
CUP1    0001 1C97   ALS-8 file.
CUP2    0001 19B8 CUP source code as ALS-8 file.
CUP2    0001 19B8
CUP3    0000 00FC CUP object code as Solos/Cuter file.
CUP3    0000 00FC
SCDO1   0001 13B2 A routine to execute a list of Solos commands on
SCDO1   0001 13B2   the screen.  Documentation and source file.
SCDO2   0001 03CF
SCDO2   0001 03CF
LIST  U 3E00 01D1 Makes a file at the beginning of your cassette and
LIST  U 3E00 01D1   containing a directory of files on the cassette.
BSHIP G 0000 1121 Battleship game
BSHIP G 0000 1121
TAPE2   C900 0095 A tape test program to record and read back a test
TAPE2   C900 0095   pattern.
PRNIN   CC00 0400 Displays instructions for PIRAN.  Press CLEAR key
PRNIN   CC00 0400   then MODE 15 times then type GE and press return key.
PIRAN G 1000 0901 Piranha game.  Outmaneuver these voracious fish and
PIRAN G 1000 0901   swim to safety while amassing points.
SS2A  F 2A00 0600 Single-line simulator.  Single Steps through a
SS2A  F 2A00 0600   machine language program, displaying registers at
SS3A  F 3A00 0600   each step.  The only difference between SS2A, SS3A,
SS3A  F 3A00 0600   ...SS6A is the load point of the program.  Each
SS4A  F 4A00 0600   version begins execution at its starting point
SS4A  F 4A00 0600   (2A00, 3A00, etc.).  Program will then want a 4 digit
SS5A  F 5A00 0600   address where simulation is to begin in your program.
SS5A  F 5A00 0600   Space bar single steps.  See complete instructions
SS6A  F 6A00 0600   in C7 Documentation.
SS6A  F 6A00 0600
BAUD    C900 00FF Object file of ASCII-to-BAUDOT output driver by Bill
BAUD    C900 00FF   Jones.  See Proteus News, Vol 2, No 2, p 11.
UTIL  U CB00 00A0 Memory fill and ASCII dump utilities by Lewis Moseley.
UTIL  U CB00 00A0   See Proteus News, Vol 2, No 2, p 19.  Object file.
MTEST U CB00 00E0 Memory test by Lewis Moseley.  See Proteus News, Vol 2
MTEST U CB00 00E0   No 2, p. 14.  Object file.
UTSYM   2000 0F35 Utility program above, in source file (ALS-8 style).
UTSYM   2000 0F35
MTSYM   2000 1586 Memory test source file.
MTSYM   2000 1586
ASSM  O F000 1000 Assembler similar to SCS16 above, based upon PTC's
ASSM  O F000 1000   SP#1.  Resides in top of memory.
TASSM O F000 1000 ASSM above, modified to assemble 300 baud cassettes
TASSM O F000 1000   from tape source to memory object.
ASSM    0100 237B Source modifications to original SP#1 to create ASSM.
ASSM    0100 237B
TASSM   0100 096D Source Modifications to ASSM to create TASSM object.
TASSM   0100 096D
DASCI   0100 05D2 Dump ASCII source code.
DASCI   0100 05D2
EODO    0100 0B34 Enter octal/dump octal, program source.
EODO    0100 0B34
```

UNCLASSFIED ADS

WANTED:
My SOL is down and I need a main power transformer and any available information.
Paul Kittle (714) 824-4585-days, (714) 796-1300-evenings.

FOR SALE:
Three ring white notebooks imprinted with "Proteus" name to store your issues - 1 1/2" rings, will hold about 250 pages of PROTEUS. Prices include shipping. One: $8.00, two: #15.50, three: $22.50, 4 or more: $7.25 each.
Bill Jenkins: 245 E. Adele, Villa Park, IL 60181

WILL DUMP LISTINGS:
You may notify your members that I have access to a standard 132 column impact printer and can dump listings from any PT product, ALS-8, 5k BASIC, EXT BASIC, FOCAL, PILOT, MICROSOFT BASIC, etc. Simply send cassette media (copy please, not original) along with sufficient return postage and $1.00 handling fee (for envelope ect.), Expect a 10 day turnaround after I receive it.

Jim Williams
58 Trumbull St., New Haven, Conn. 06510

FOR SALE--Helios II System plus Okidata 110 printer (RS232 and tractor) and a Sanyo monitor. All low mileage. Includes SOL Rev. D with 48K memory, PTDOS 1.5(mod 2), serial port driver for printer and some games. All offers will be considered. $2500. without printer, make offer on printer.
Rod Lee, 5 West Creek Court, Lafayette CA 94549, (415)835-9566

SOL FOR SALE:
Clean SOL Rev. #, 16 Kra, with all documentation, programs and manuals. $900.
Millard Edgerton, (415) 948-3818 day or evening.
(days) or (415)284-7278 (home).

WANTED: Used Helios Disk system or perhaps other disk system that will interface with my SOL w/ 2708 Personality Module (48k Memory). Also a low cost printer.
Clarice Turner, 874 Sunset Dr., Livermore, CA 94550

FOR SALE, SOL/20 latest rev. R, complete
with Solos and 64K Expandoram $ 750.00.
B. Diller 213-9666020

FOR SALE - SOL-10 Computer with SOLOS in PROM. Includes SOL-20 backplane board for upgrade to SOL-20, if desired, $550. Call Kevin Counselman at (205) 883-4291 after 5 CST or write to 11304 Hillwood Drive, Huntsville, AL 35803

FOR SALE: 8" HARD-SECTOR DISKETTES (Suitable for use in HELIOS Systems). New, house-brand (ie, no-names). Sold in lots of 10 only. In hard plastic library case, $30.00/box of 10. Without box, $26.00/10. Prepaid orders shipped free within the continental United States (elsewhere, add extra for shipping as appropriate.) Will ship C.O.D. in U.S. with $5.00 deposit plus you pay COD fees and actual shipping.

Order from:
Lewis Moseley, Jr.
2576 Glendale Court, N.E.
Conyers, Georgia 30208

WANTED

Tutor for BASIC Programming.
Familiarity with Processor Technology
SOL MOST helpful

CALL -
(212) 242 4521
Evenings or Early AM
Glen R. Snider

Updates are available for the following PTDOS programs:

WordWizard 4.0.2 System Disk.
WordWizard Document Disk, Rev. D.
ACCPAC Accounts Receivable Daily Processing System Disk 1.0.1.
ACCPAC Accounts Receivable Monthly Processing System Disk 1.0.1
ACCPAC Programmers Package 1.1.0 System Disk.
ACCPAC General Ledger 1.3.4 System Disk.
ACCPAC Financial Report System Disk 1.1.2.
ACCPAC Universal Co. General Ledger Data Disk, Rev. A.
ACCPAC Universal Co. General Ledger Report Documents, Rev. A.
ACCPAC Accounts Payable Daily Processing System Disk 1.0.2.
ACCPAC Accounts Payable Monthly Processing System Disk 1.0.3.
MailMaster System Disk, 3.0.2.
MailMaster List Disk, 3.0.2.
MailSort System Disk, 1.0.1.

Send your original Processor Tech disk to PROTEUS, 1690 Woodside Rd. Redwood City, CA 94061, Attn: Jane, with a check for $10.00 for each disk to be updated. Please allow a week turnaround time.

TABLE OF CONTENTS

STOLEN EQUIPMENT ALERT!

The Computer Port (Arlington, Texas) has issued an alert to be aware of Sol computers, Helios disk drives, and PerSci floppy disk drives that were stolen in a burglary there. These items were malfunctioning units in the shop for repair, so service centers should be aware of the serial numbers as the theives may try to have them fixed. Buyers of second hand equipment should also be on the lookout. If a stolen unit is identified, please call your local police immediately and also call the Computer Port collect for instructions. Telephone (817) 469-1502. Here's the full list of stolen Processor Tech products:

```
        PER SCI DISK DRIVE, SERIAL #'S  4381, 2382, 4526,
                                        4332, 2695, 4488.
        SOL-20 COMPUTER, SERIAL #'S 400742, 401139.
        HELIOS II DRIVE, SERIAL # 501763.
```

PROTEUS/NEWS

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

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

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

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

18104

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

September/October
November/December 1981

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

Vol. 4 #5/6

Single Issue $4. (US)
Double Issue $8. (US)

UPGRADE SOL TO 24X80 SCREEN

In previous issues, we have reported that Micro Complex in Southern California was working on a series of upgrades for the Sol to make it more compatible with new microcomputers. One such upgrade was the Dual Personality Module that allowed the Sol's dedicated address space to be switched from C000 to F000 to give 60K contiguous memory below it. Now we have another series of improvements, and more are to come.

The new product is called the Micro Complex 80/64 module. This upgrade consists of a circuit board that is fully assembled and tested, but requires some work to install it on your Sol's main board. You need to remove 30 IC's from the front-left corner of the Sol (under the keyboard area), solder a few wires to various points on the Sol, disengage a few pins from other IC's on the Sol, and plug the modification board, piggy-back-style, into some of the vacated IC sockets.

FEATURES

The new modification does a few things. It modifies the video display to the industry standard 24x80 size, it improves the quality of the screen image by eliminated jitter and snow, it allows the dedicated RAM and ROM in the Sol to vanish from the memory space so 64K RAM can be put into the S-100 bus, it generates a tone when the ASCII bell character is recognized, and it speeds up the on-board RAM circuitry so that the Sol can be run with a faster clock for the Z-80 modification that Micro Complex is also developing.

The video display can be switched between a 16 line by 64 column format (the present Sol display) or a 24 line by 80 column format. In the 80 column format, a portion of the Sol's extra system RAM is used as the additional video RAM. The mapping of memory to screen location is line by line, as you would expect.

To select the format desired, you use the switch on the Micro Complex Dual Personality Module. That is, when the personality module is in the standard mode (C000 origin), the screen is standard 16x64 size. When you switch to the extended mode (F000 origin) you automatically switch the screen size, too. The Solos program in the personality module has been modified to operate the larger screen.

Although not explained in the documentation, it seems possible to use the 80/64 without using the Dual Personality module, but modification of the program in the Solos ROM is required, as well as some minor hardware changes to the personality module. This would necessitate using one module for the 80 column mode and another when you want to go back to the present 64 column mode. This change has been described in previous issues of Proteus News in connection with the "move Sol from C000 to F000" modification.

The 80/64 module contains a connection to an audio tone generator. When the ASCII bell character (07H) is recognized, the tone is generated. This is useful for some software which uses the bell character to alert the operator to an unusual condition, such as overflowing input beyond the expected field size. If you have software using the bell character this way, what you see on a standard Sol is a "little man" (actually a little bell symbol) instead of hearing the bell. For example, PTDOS gives this when you backspace at the beginning of a command line or enter more than 80 characters on a command line. With the 80/64, this should give a tone.

The Solos monitor and the Sol RAM (video & scratchpad) can be made to disappear from the address space, so that 64K RAM can be plugged into the S-100 bus and fully utilized. The disabling is done under software control, by outputting 1's to the two high order bits of port FC. When your operating system wants to use the screen or Solos, it should output 0's to these port bits.

Two extra port bits are available as software controlled flip-flops that may be used by your custom circuitry. One application suggested in the manual is connecting one flip-flop to the disable line of a memory-mapped disk controller (such as NorthStar) to get it out of the address space when it is not being accessed.

The 80/64 module also has changed the manner in which the video RAM is accessed by the hardware signals. It used to respond to the S-100 bus, but it is now located on the internal bus. This is why memory in the S-100 bus can overlap the video RAM without conflict. This is also said to eliminate the horizontal jitter which occurs in a normal Sol.

Not only is jitter eliminated, so is the streaking which occurs when the screen is repeatedly accessed by the program. In the standard Sol, when a program (either Solos or a user's program) reads or writes the video RAM, the video beam is momentarily turned off while the RAM is in use. This results in a short blank streak on the screen. To see this streak on your present Sol, hold down the repeat key and a character key and carefully watch the screen for short horizontal streaks that randomly flash on the screen. In some video games, where the screen is rapidly updated, the streaking is very obvious. With the 80/64 modification installed, the screen does not streak.

Hardware scrolling is still preserved, but the "windowshade" feature had to be eliminated to allow the extra lines of display to be scrolled. No one used windowshade anyway, so this is no loss.

The Sol's built-in RAM, which is used for video display and Solos scratchpad, has been replaced by higher speed RAM on the 80/64 board. A Hitachi 6116 P-3 static 2K RAM chip is

CON'T FROM PAGE 1-UPGRADE SOL TO 24 X 80 SCREEN

used. This is the new generation of RAM used on those ultra-low power RAM boards that allow RAM or 2716 ROM IC's to be plugged in interchangeably. The 150 ns access time of this chip will allow the Sol to operate with a 3.57 MHz clock when used with the MicroComplex Z-80 modification that is under development. The on-board wait state generated by Sol can be eliminated so that the Tarbell disk controller and other disk controllers can be bootloaded directly from the personality module.

IMPROVED PERSONALITY MODULE

Micro Complex can provide personality ROM's customized so that your disk controller will boot on a built-in command or single keystroke when in Solos. The Dual Personality Module can be ordered with NorthStar boot, Tarbell boot, or Helios boot as standard. It can also be ordered to support a Tarbell/Helios multiplexer setup. Custom versions can be arranged on special order.

The Solos in the personality module that Micro Complex supplies has been modified by elimination of the cassette routines. Since most systems are using floppy disks now, the cassette routines are seldom used and can be loaded from disk when necessary. (You'll have to program them onto disk from a listing of the original Solos. CP/M users group library has a Solos source file available on one of the library disks.) Or you can switch back to the old personality module when you need to manipulate cassette files, such as when using DISKTAPE and TAPEDISK utilities developed by Greenlaw.

In place of the cassette routines, you have numerous new commands that are useful for probing around in machine language, such as ASCII dump, hex dump, enter ASCII, enter hex, fill memory with a hex byte, hex to decimal conversion, move memory block, compare memory blocks, address reference locator, find (ASCII, byte, word), and memory test.

Since Micro Complex also sells a work-alike NorthStar-type disk controller of their own design, the personality module supports features especially useful for NorthStar disk owners. Commands are provided to jump to various entry points in NorthStar BASIC and DOS, as well as for warm-starting CP/M. The personality module is available without the 80/64 module.

HOW HARD IS IT TO INSTALL?

The recommended procedure for installing the 80/64 piggy-back board involves removing the Sol's main board from the Sol cabinet. This means that the keyboard is removed, the S-100 bus backplane is removed, the internal cables are unplugged, and the Sol PC board is dismounted by removing the screws underneath the Sol base. It is possible to install the board with the Sol PC still in place, but it is more difficult and not the best way.

With the Sol PC on the bench, the designated IC's are lifted out of their sockets. Other IC's are removed to allow certain pins to be bent outward partially, and then they are reinserted into their sockets with the bent pins out of the socket. A few wires from a ribbon cable socket provided with the 80/64 board are soldered to designated points on the Sol.

The edge connector which holds the personality module needs to have two of its pin connectors unsoldered and lifted out of the Sol PC holes for soldering to the ribbon cable. By cutting two traces on your old personality module, you can still use it after the alterations are made, although you will usually be using the new Dual Personality Module.

One or two jumpers are soldered to the underside of the Sol PC or to IC pins. All of the changes to the Sol PC are easily removed, so if you ever want to go back to the way your Sol used to be, it can be done. I don't think you'll ever want to, though.

After the Sol PC is prepared, the 80/64 board is carefully aligned so that precisely aligned, long, gold-plated pins extending down from it will fit into some of the empty IC sockets on the Sol PC. The board is pressed into place, the pins are checked to be sure they all went into place, the ribbon cable is plugged into the 80/64 connector, another ribbon cable is plugged into a Sol PC socket, and so on. The installation is not hard, but requires attention to detail. Micro Complex has made it as neat and professional as possible.

After the connections are all completed, the Dual Personality module is plugged into the Sol PC instead of your old module. The internal cables are re-connected and the Sol is powered-up to test it. After successful check-out, the screws and S-100 backplane are re-installed, the system is re-checked, and you're done. It can be accomplished in one afternoon.

HOW WELL DOES IT WORK?

Micro Complex has sent out a number of modules for field testing. Proteus is in the process of using one, but there are some bugs yet to be eliminated before I can give a full report on the features. This article is being written on my Sol/Helios system with the 80/64 module installed, so I can assure you that the module does exist and does work in the 64 column mode. I am having some problems with Solos commands and some occasional bugs with the video display, so I am going to send my Sol PC with the 80/64 to Micro Complex for checkout and debugging. In the next issue, I'll report the outcome.

I have switched the system to the 80 column mode, but I can't fully try that because of the bugs I am having. The system crashes erratically when using Solos, but I am able to bootload using the bootload command built into the Helios version of Solos. I have no crashes when running programs out of the S-100 bus RAM. DMA into and out of the S-100 bus works fine. The problems happen when I use the Solos ROM and built-in RAM. The Sol I have is an old Rev D version, so I suspect that there is some problem with the old Sol that the new Sol doesn't have, there is a critical timing or noise problem in the modification, or I have a poor connection somewhere.

I can say, though, that the 80 column screen image is beautiful. Using the BMC video monitor model KG-12C, which is the one recommended and sold by Micro Complex, the characters are sharp from corner to corner of the screen. There is no trouble reading the characters at the edges and the image fits nicely on the screen. There is no snow or jitter at all, in either 64 or 80 mode. It is a very professional looking screen.

In the 80 column mode, there is some barely noticeable flicker, but this can be reduced by a jumper that is explained in the 80/64 instructions. This jumper eliminates the blank scan-line that the Sol normally puts between the descender of a character (e.g., the bottom hook of a lower case "y") and the top of the character below it. This mod reduces the time required to refresh the screen, and thus cuts down on the flicker. I have hooked up this jumper through the unused switch #2 on the DIP switch at the front of the Sol, so I can cut the modification in or out. This BMC monitor has a green phosphor with an extended persistence, so the flicker is just barely noticable, and it will bother few users. With the modification switched in, the flicker almost is undetectable to my eyes. In the 64 column mode, there is no flicker at all.

CON'T FROM PAGE 2-UPGRADE SOL TO 24 X 80 SCREEN

I get a spurious character on the video screen at rare times when I am using my wordprocessor (WordWizard for Helios). That is, once in a while, when I press the REPEAT and DEL keys to delete a bunch of characters, or when I move the cursor, I get a parenthesis appearing on the line. I know it is not a keyboard problem, and suspect that there is some subtle timing or noise problem when a program writes to the screen RAM in my system. I'll let you know what Micro Complex discovers.

PRICE AND AVAILABILITY

Micro Complex has set the price of the 80/64 modification at approximately $250. You also need the Dual Personality Module at approximately $95. Shipping extra. Be sure to specify which disk controller you use, so the correct version of Solos can be supplied.

A higher resolution video monitor is needed for 80 columns than you may be using now for 64 columns. The recommended BMC monitor sells for about $225. It needs a custom video cable to mate with Sol's video connector, which you could make yourself with parts from Radio Shack, or you can order one from Micro Complex for $10.

Considering the fact that any S-100 video display board you can get will probably cost more than $250 and will not have any of the custom Sol features of the 80/64 board, this is really a reasonable price.

Micro Complex will install and test the 80/64 modification for you in their shop for $50 plus shipping. They can also repair non-functional Sol's. Ask them for proper shipping instructions.

FURTHER INFORMATION

Contact Robert Hogg at Micro Complex, 25651 Minos Street, Mission Viejo, CA 92691, telephone (714) 770-2168.

....See last minute P.S. on page 15....

PUBLIC DOMAIN GENERAL LEDGER FOR N*
by Franz J. Hirner

Having recieved a lot of useful information from PROTEUS over the past two years, I thought I should sit down and contribute something that other members may find useful.

Eclosed please find a draft of a article describing GENERAL LEDGER which I wish to offer for ther possible use of other members.

I have been using the GENERAL LEDGER System as presented for two years to keep books for two small business applications as well as my personal finances. While there are more complete packages available which integrate Payroll, Accounts Receiveables, etc. from several vendors, I have not seen a flexible system that can be implemented by the novice user as easily as GENERAL LEDGER and still provide the useful reports necessary to control small business operations.

Due to the length of the programs many members may not want to take the time to type each program. I would provide disk copies of the programs as presented here in North*Star Duble Density format plus a copy of the description of the system for $25.00.

Sincerely,

Franz J. Hirner

631 Matsonia Drive
Foster City, California 94404
November 27, 1981

GENERAL LEDGER

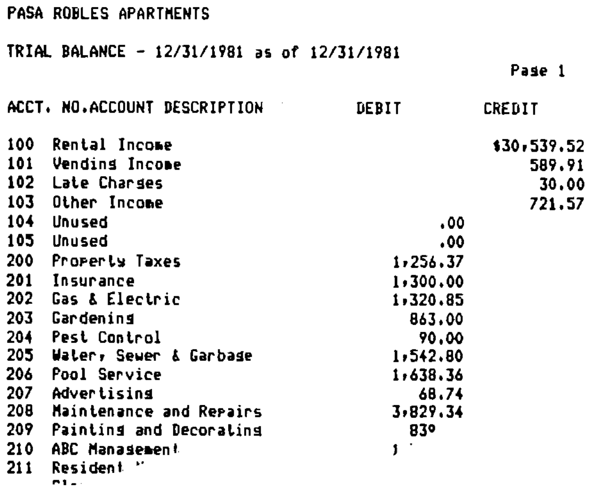
The General Ledger package is written in North*Star Basic. GENERAL LEDGER should operate under both North*Star Single Density and Double Density Disk Systems. With user modification GENERAL LEDGER should be able to run under most versions of Basic.

General Ledger ia a simple Ledger Package compared to many systems on the market but should be adequate for a wide variety of small business and personal applications. GENERAL LEDGER does contain useful features not found on some systems such as monthly budget comparisons, year to date budget comparisons, etc.

GENERALLEDGER, as presented here, has been implemented on a Sol-20 Computer with 48K memory, dual North*Star Double Density Disk System. The programs are written in North*Star Basic. The only program code unique to the system is the use of Control-(11) to clear the screen as required for the Processor Technology Sol-20 Computer System. The user will have to change to the appropriate code to match the requirements of his terminal. All other code is standard North*Star Basic and should run on any computer equipped with North *Star Disk Drives and North*Star Basic without modification.

The GENERAL LEDGER System programs can be classified into two types of programs: 1.) those that are necessary to initialize the data disk and create the proper ledger files and 2.) those that are used to enter transactions, update files, and produce the various reports. A Single Density North*Star System will require that those programs necessary to set up the files be kept on a separate disk due to space limit ations. Users with Dual Density Systems may also want to do the same for data security reasons.

SAMPLE OUTPUT FOLLOWS:



```
PASA ROBLES APARTMENTS

TRIAL BALANCE - 12/31/1981 as of 12/31/1981
                                                   Pase 1

ACCT. NO.ACCOUNT DESCRIPTION         DEBIT         CREDIT

100  Rental Income                               $30,539.52
101  Vendins Income                                  589.91
102  Late Charses                                     30.00
103  Other Income                                    721.57
104  Unused                               .00
105  Unused                               .00
200  Property Taxes                  1,256.37
201  Insurance                       1,300.00
202  Gas & Electric                  1,320.85
203  Gardenins                         863.00
204  Pest Control                       90.00
205  Water, Sewer & Garbase          1,542.80
206  Pool Service                    1,638.36
207  Advertisins                        68.74
208  Maintenance and Repairs         3,829.34
209  Paintins and Decoratins           83°
210  ABC Manasement                  )
211  Resident
```

CON'T FROM PAGE 3-PUBLIC DOMAIN GENERAL LEDGER FOR N*

```
PASA ROBLES APARTMENTS                  BUDGET STATEMENT              PERIOD ENDING 12/31/1981

          CURRENT PERIOD                                                  YEAR TO DATE
  BUDGET     ACTUAL   VARIANCE     % ACT  ACCOUNT                   BUDGET     ACTUAL   VARIANCE    %

INCOME

 2395.00    2475.00      80.00      3  100 Rental Income           28740.00   30539.52   1799.52     6
   48.40      54.63       6.23     13  101 Vending Income            580.80     589.91      9.11     2
    1.00        .00      -1.00   -100  102 Late Charges               12.00      30.00     18.00   150
    4.08        .00      -4.08   -100  103 Other Income               48.96     721.57    672.61  1374
     .00        .00        .00      0  104 Unused                       .00        .00       .00     0
     .00        .00        .00      0  105 Unused                       .00        .00       .00     0

 2448.48    2529.63      81.15      3      TOTAL INCOME            29381.76   31881.00   2499.24     9


EXPENSES

  170.78    1256.37    1085.59    636  200 Property Taxes           2049.36    1256.37   -792.99   -39
  107.41        .00    -107.41   -100  201 Insurance                1288.92    1300.00     11.08     1
   86.32      92.50       6.18      7  202 Gas & Electric           1035.84    1320.85    285.01    28
   83.75      65.00     -18.75    -22  203 Gardening                1005.00     863.00   -142.00   -14
    7.50       7.50        .00      0  204 Pest Control               90.00      90.00       .00     0
  135.28     127.38      -7.90     -6  205 Water, Sewer & Garbage   1623.36    1542.80    -80.56    -5
  169.76     125.00     -44.76    -26  206 Pool Service             2037.12    1638.36   -398.76   -20
   23.15        .00     -23.15   -100  207 Advertising               277.80      68.74   -209.06   -75
  392.69     372.60     -20.09     -5  208 Maintenance and Repairs  4712.28    3829.34   -882.94   -19
   49.13      25.00     -24.13    -49  209 Painting and Decorating   589.56     839.13    249.57
  116.25     125.00       8.75      8  210 ABC Management           1395.00    1711.07    316.07
  100.00     100.00        .00      0  211 Resident Manager         1200.00     815.00   -385
    8.00        .00      -8.00   -100  212 Cleaning Exp                           68.00
                           .00      0  213 Legal Ex
                        -25.93   -100  214 Misc.
                                   -3  215 In
```

```
GENERAL LEDGER JOURNAL - 12/31/1981                          Page 1

JOURNAL NO. 13

TRANS  ACT. NO.  DESCRIPTION                 ENTRY   DATE           DEBIT          CREDIT


TRANSACTION: INCOME - RECIEPTS 178 TO 188

 1      100      Rental Income                  1  12/31/1981                    2475.00
        300      Cash & Checking Accounts       2  12/31/1981    2475.00

TRANSACTION: VENDING - AAA VENDING

 2      101      Vending Income                 3  12/31/1981                      54.63
        300      Cash & Checking Accounts       4  12/31/1981      54.63

TRANSACTION: SACTO COUNTY - 1981 PROPERTY TAXES

 3      200      Property Taxes                 5  12/31/1981    1256.37
        300      Cash & Checking Accounts       6  12/31/1981                    1256.37
```

```
TRANSACTION: EXPENSEX - CHECKS 187 TO 195

 4      202      Gas & Electric                 7  12/31/1981      92.50
        203      Gardening                      8  12/31/1981      65.00
        204      Pest Control                   9  12/31/1981       7.50
        205      Water, Sewer & Garbage        10  12/31/1981     127.38
        206      Pool Service                  11  12/31/1981     125.00
        208      Maintenance and Repairs       12  12/31/1981     372.60
        209      Painting and Decorating       13  12/31/1981      25.00
        210      ABC Management                14  12/31/1981     125.00
        211      Resident Manager              15  12/31/1981     100.00
        300      Cash & Checking Accounts      16  12/31/1981                    1039.98

TRANSACTION: MORTGAGE PAYMENT - FIRST SAVINGS & LOAN

 5      300      Cash & Checking Accounts      17  12/31/1981                    1152.00
        600      Mortgage Payable              18  12/31/1981      27.23
        215      Interest Expense              19  12/31/1981    1124.77

TRANSACTION: POST NOVEMBER 81 PAYMENT - ENTERED

 6      500      Accounts Payable              20  12/31/1981                    1260.91
        300      Cash & Checking Accounts      21  12/31/1981    1260.91

                                       JOURNAL TOTAL            7,238.89        7,238.89
```

```
FINANCIAL STATEMENT - PERIOD ENDING 12/31/1981              Page 1

CURRENT ASSETS

Cash & Checking Accounts                   $3,843.57      $3,843.57


FIXED ASSETS

Land                                      $20,406.84
Buildings                                 147,048.99
  LESS: Accu. Depr. - Buildings          -27,325.14
Furniture                                   3,401.14
  LESS: Accu. Depr. - Furniture           -3,345.35
Excess Market/Book Value                   96,009.75    $236,196.23
                TOTAL ASSETS                            $240,039.80


CURRENT LIABILITIES

Cleaning Deposits                            $270.00        $270.00


LONG TERM LIABILITIES

Mortgage Payable                         $135,246.63    $135,246.63
                TOTAL LIABILITIES                       $135,516.63


CAPITAL

Equity                                   $107,009.08
PROFIT OR LOSS                             -2,485.91    $104,523.17
                TOTAL CAPITAL AND LIABILITIES           $240,039.80
```

CON'T FROM PAGE 4-PUBLIC DOMAIN GENERAL LEDGER FOR N*

```
                                        PASA ROBLES APARTMENTS

PREPARED 12/31/1981 AS OF 12/31/1981           GENERAL LEDGER TRIAL BALANCE                Page 1

      $$$$$$$$ TRANSACTION ACTIVITY $$$$$$$$$      $$$$$$$$ HISTORY BALANCE $$$$$$$$$$
ACT   JNL ENTRY   DATE        DEBITS     CREDITS       OPENING      NET CHANGE     CLOSING


100 Rental Income
      13    1   12/31/81               2,475.00
        CURRENT                .00     2,475.00     28,064.52CR    2,475.00CR    30,539.52CR


101 Vending Income
      13    3   12/31/81                  54.63
        CURRENT                .00        54.63        535.28CR       54.63CR       589.91CR
```

```
                                        PASA ROBLES APARTMENTS

PREPARED 12/31/1981 AS OF 12/31/1981           GENERAL LEDGER TRIAL BALANCE                Page 2

      $$$$$$$$ TRANSACTION ACTIVITY $$$$$$$$$      $$$$$$$$ HISTORY BALANCE $$$$$$$$$$
ACT   JNL ENTRY   DATE        DEBITS     CREDITS       OPENING      NET CHANGE     CLOSING


210 ABC Management
      13   14   12/31/81      125.00
        CURRENT               125.00          .00       1,586.07        125.00       1,711.07


211 Resident Manager
      13   15   12/31/81      100.00
        CURRENT               100.00          .00         715.00        100.00         815.00


215 Interest Expense
      13   19   12/31/81    1,124.77
        CURRENT             1,124.77          .00      11,884.06      1,124.77      13,008.83


300 Cash & Checking Accounts
      13    2   12/31/81    2,475.00
      13    4   12/31/81       54.63
      13    6   12/31/81               1,256.37
      13   16   12/31/81               1,039.98
      13   17   12/31/81               1,152.00
      13   21   12/31/81    1,260.91
```

```
                   INCOME STATEMENT

 FOR PERIOD ENDING 12/31/1981                         Page 1

 INCOME

 Rental Income                     $30,539.52
 Vending Income                        589.91
 Late Charges                           30.00
 Other Income                          721.57

         TOTAL INCOME                            $31,881.00
```

```
EXPENSES

Property Taxes                         $1,256.37
Insurance                               1,300.00
Gas & Electric                          1,320.85
Gardening                                 863.00
Pest Control                               90.00
Water, Sewer & Garbage                  1,542.80
Pool Service                            1,638.36
Advertising                                68.74
Maintenance and Repairs                 3,829.34
Painting and Decorating                   839.13
ABC Management                          1,711.07
Resident Manager                          815.00
Cleaning Expense                           68.00
Legal Expense                             243.00
Misc./Other Expense                       302.50
Interest Expense                       13,008.83
Depreciation - Buildings                5,205.39
Depreciation - Furniture                  264.53

          TOTAL EXPENSES                              $34,366.91


                          LOSS                        $-2,485.91
```

Franz J. Hirner
631 Matsonia Drive
Foster City, CA 94404

Helios/Morrow Disk Multiplexer Allows Hard/Soft Sectoring

Ace Computers, Inc. has told Proteus that they have a prototype of a board that allows the Sol with Helios disk to use either Helios format diskettes, or standard 8" soft sectored diskettes. A Morrow floppy disk controller must be plugged into the Sol bus, as well as the Helios board set. Both ribbon cables are plugged into the back of the Helios cabinet. Inside the Helios, the Ace board plugs between the PerSci drive and the ribbon cables.

When a floppy disk is inserted, the Ace board detects whether it is a hard-sectored or soft-sectored diskette, and switches to the appropriate controller, more or less. This will allow Helios users to continue to use existing PTDOS software and also to use standard CP/M soft sectored diskettes with CP/M. (CP/M 2.2 for the Morrow board is included with purchase of Morrow's controller.) They say it preserves the fast-seek feature of the PerSci drives, something which many floppy disk controllers won't do.

The Ace people were going to demonstrate this board to me, but illness and car breakdowns have so far interfered with scheduling this. When I see it in action, I'll let you know the results.

Ace, you may recall, is the service company formed by two former PTC engineers. They do hardware repairs of PTC products, including top-notch Helios maintenance, and repairs on other S-100 microcomputer products such as NorthStar and Morrow.

USER REACTS TO 24 X 80 CONVERSION
by Earl Dunham

Stan;

I think this news is important and I hope you can get this report in the very next PROTEUS. You probably have the conversion yourself so you know what I am talking about.

I was at Bob Hoggs house the night he called you about the mod. he had sent you. If you have not met Bob, make it a point to do so; he is everything I said he was in my previous report and more so. He will be the saviour of SOL I'm sure. Your contributions are not at all diminished by his efforts; I appreciate them and I think he does too. If you two combine your efforts, each in your special way, we shall truly retell the story of Phoenix.

Working with Bob is an experience to cherish his skill and knowledge is overwhelming. I would not part with any of my SOLs, in fact I may buy every one I can find at the current bargain prices they are going for. Those who abandon SOL will regret it someday.

I will send you a lengthy report in a few weeks when I have more detailed information about the conversion. I am really anxious to get the Z80 mod running; I'll let you know

Earl Dunham

RUN!! DONT WALK!!
The SOLution is here!

Dear Stan;

I'm sure by now that you are aware of the exciting new miracle from the work shop of Bob Hogg, the great technician who gave us the Dual Personality Module that I reported in the last Proteus. Now, as promised, he gives us the 80x24 screen with his Dual 80/64 modification. I had the modification done last week and I want to tell everyone what I think about it.

Sensational, Terriffic, Unbelievable!! I could go on and on with the superlatives, but I'm sure you get the idea that I am absolutely delighted; no, more than that, a great deal more; I am ecstatic. I was not really aware of the severe limitations imposed by the 16x64 screen. Even on something as simple as a code listing, the 24 lines makes debugging much easier. It is as though we have been looking through a very small port-hole and have suddenly been allowed on deck for the first time. The view is incredilble. I will only spend a little time on the technical aspects of the conversion for this report, just enough so that everyone can be aware of what the SOLution involves. Make no mistake SOLdiers, anyone who uses SOL in a serious way just must have this improvement.

The conversion involves removing about two dozen I. C. s, mostly the on board ram between U1 and U33, soldering in several jumpers, and the installation of a piggy-back board in the area on the left, just below the keyboard. The board is unique. It looks like a miniature, inverted bed of nails. It has a veritable forest of inch or so long pins poking out the bottom. These pins are spaced so that when the board is positioned correctly they snap into appropriate pin holes vacated by the removal mentioned above. There is a single 2K static ram and about a half dozen other chips on the board, and a couple of edge connectors.

The Dual 80/64 modification is offered as a kit or Bob will do the installation for a modest fee. It is my impression, although Bob can do it in an hour or so if there are no problems, that the week-end bit fiddler will require much longer. But remember, Bob designed the thing and has alredy made several installations. I do not mean to imply that it is terribly difficult; it just aint a walk in the park.

The installation can be made without removing the motherboard from SOL; but it shouldn't. It is too difficult to see and the positioning of the pin over the holes must be just right when the piggy-back is pressed home. Because considerable pressure is required, the wrong pins in the wrong holes would be a disaster; bent pins would cause a lot of headaches. Most of us have dissasembled SOL enough times that it aint no big deal anyway. Besides, it gives us a chance to clean out all the debris that has sifted in.

Finally, you must have the Dual Personality module and a monitor of sufficient bandwidth to handle nearly twice as many characters on the screen as before. Bob has found and sells a P31-Green Phosphor 12 inch professional monitor with an 18 Mhz bandwidth made by BMC (model KG-12c). It costs a little over $200 and is superior to any I have seen. Filled corner to corner, the resolution is terrific. The screen remains jet-black and the contrast makes the charcters just jump out at you. It comes with an anti-glare screen that really works. Consider it even if you dont make the conversion. But I cannot imagine anyone not doing whatever is neccesarry to get it. Bob adds a little buzzer that gives a pleasant little bleep on power-up and responds to 07H. A nice little touch. Use it as the final reason why you cannot live without the wide screen.

Now SOL with its built in quality and great design is back in the ball game. In a few weeks Bob Hogg will have the Z80A conversion ready and then look out. Apples and Pets and Trash-80s will be Mickey Mouse compared to SOL running at 3.57 Mhz with an 80x24 screen. Hard-disks are under developement for SOL in several places also. The future looks bright for we who have suffered the slings and arrows etc. for so long.

You ask what will it cost? Buying everything required from Bob and letting him do the installation will create a bill of something over $500 (plus shipping of course). I would rather not try to be specific; a phone call to Bob will get you the straight dope. It will be the best $500 you have spent since you got your computer.

Bob Hogg is:

Micro Complex
25651 Minos St.
Mission Viejo, CA
92691

Happy Days;

Earl Dunham

Ph:(714) 770-2168

A RESPONSE TO DESLAR K. PATTEN'S OPEN LETTER

By Allen T. Fincher

This article is in response to Deslar Patten's letter in vol. 4, no. 3/4 issue concerning the SOLOS "clear-to-end-of-line" feature.

First a little background. Most video terminals around also have a "clear-to-end-of-line" feature built in them. Since the normal line termination sequence is a carriage return followed by a line feed, the terminal processes the CR as expected, by returning the cursor to the left margin of the screen. But, the LF character actually causes two operations to occur. First, the cursor is moved down to the next line (or a scroll is performed), and then the new line is cleared from the current cursor position to the right margin.

In SOLOS though, the internal termination sequence is LF/CR (locations C2F9 thru C302). Therefore, the VDM driver software performs a clear-to-end-of-line on CR, not LF giving the effect commented on by Deslar. The only reason I can determine that this was done is based on the command line termination options in SOLOS. Namely, a CR terminates the command erasing all characters to the right of the cursor, and a LF just terminates the command. This would allow the operator to use the cursor-left key to place the cursor over an incorrect letter in a command, correct it, and then press LF to process it instead of using DEL and retyping the rest of the command over. This can be done because SOLOS actually uses the current line on the VDM screen as the command-line buffer. So that after either CR or LF is pressed, the current screen line is searched and read by the individual command processing routines with a space being the command terminator (a cleared line is actually 64 spaces).

The following routine can be used by a Sol owner who does not have the ability to reassemble SOLOS. It can reside anywhere there is free memory. The origin shown is in the User area at CAB4H in the SOLOS System Global Area ram. This routine performs the second type of operation that Deslar gave in his letter. That is, any carriage return that immediately follows a carriage return, is not sent to the VDM display driver.

After assembling the routine, load the object code into ram (it should load at the ORG'ed addresses). If the last ORG and DB 03H is omitted, perform a "O=3" command to enable your new preprocessor. Note that this routine is not used when you are in SOLOS, but only from external programs.

Please note that all addresses given in this article are for SOLOS Version 1.3, Release 77-03-27 as printed in vol. 1, no. 3 June 1977 of ACCESS (Processor Technology's newsletter), and may not be the same as your monitor's version. By the way, this version of SOLOS has an error in it. The instruction at address C5E4 should be C2 FB C5 (JNZ TERE1) instead of C2 FA C5 (JNZ TERE2). The mistake will prevent the cassette versions of BASIC from displaying an error message if an attempt is made to open a file after it has already been opened. For those with source listings, this is in the BOPEN routine.

I must make a disclaimer here. Since I don't have a Sol and my present monitor, though incorporating CUTER, has had the carriage return and line feed VDM driver routines changed to cause a clear-to-end-of-line operation to occur on a LF, I cannot guarantee that the following routine will work with all external software but because a carriage return that follows another carriage return doesn't actually accomplish anything, so I really don't foresee any problems.

```
;
; THIS INTERFACE ROUTINE WILL ASSURE THAT A CARRIAGE-RETURN
; IS FOLLOWED BY A LINE-FEED.  IF A CR FOLLOWS A CR, THE
; SECOND ONE WILL BE IGNORED.
;
UOPRT  EQU  0C802H  ;User defined output routine vector location
OPORT  EQU  0C807H  ;Pseudo output port
USARE  EQU  0CAB4H  ;User ram area
;
CR     EQU  0DH     ;Carriage return
LF     EQU  0AH     ;Line feed
;
;
; Set pseudo output port vector to point to our preprocessor.
;
       ORG  UPORT
;
       DW   USARE
;
;
; The actual preprocessor.
;
       ORG  USARE
;
PREPR: MOV  A,B     ;Get the output character
       CPI  CR      ;Is is a carriage return?
       JNZ  CHOUT   ;No, display it
       LDA  CRFLG   ;Get last character
       CMP  B       ;Was it a CR also?
       MVI  A,0     ;Simulate SOLOS just in case it is
       RZ           ;Yes it was, ignore it
;
; Use the VDM (pseudo port 0) as the actual output console.
;
CHOUT: STA CRFLG    ;Save character for next time
       XRA  A       ;Cheap zero
       CALL AOUT    ;Display character in B reg. on VDM
       RET
;
; Storage for previous character sent to VDM.
;
CRFLG: DB   0       ;Initialize with 0
;
;
; This could be done with the Set Output Port command
; (O=port).
;
       ORG  OPORT
;
       DB   03H     ;User defined output
;
       END
```

Allen T. Fincher
  Suffolk, Va.
 Sept. 8, 1981

SEPT 15, 1981

STILL MORE ON THE CPM USER AREA

I'VE BEEN ENJOYING USING DR. JIM BYRAM'S SOLUSER9 SET OF CPM USER AREA ROUTINES WITH MY SOL-N* SYSTEM. I USE LIFEBOAT'S N* VERSION OF CPM AND I'VE HAD ONE DIFFICULTY WITH SOLUSER9. WHEN I WRITE A PROGRAM USING THE EDITOR (ED) THE DELETE CHARACTER ROUTINE DOESN'T DESTROY THE CHARACTER AS DESIRED. DOES THIS SOUND FAMILIAR TO ANYONE OUT THERE?

FOR REASONS NOT KNOWN TO ME IN THE INSTANCE MENTIONED ABOVE, DURING THE DELETION OF A CHARACTER AN ADDITIONAL PASS OCCURS THROUGH THE OUTPUT AREA OF SOLUSER9. NOW IF ONE DOES NOT CALL ON THE DELETION ROUTINE, THE TWO CASES (SINGLE PASS OR DOUBLE PASS) RESULT IN OUTPUTING TO THE SCREEN IN THE SAME MANNER; THAT IS, A SINGLE ECHO APPEARS WHEN THE DELETE BUTTON IS PUSHED. HOWEVER, WHEN THE DELETE ROUTINE IN SOLUSER9 REACTS WITH THE DOUBLE PASS CASE THE ECHO REMAINS UNDESTROYED. THE PATCH THAT I HAVE DEVELOPED HANDLES THE TWO CASES. I'VE INCLUDED THIS PATCH TO SOLUSER9 FOR THOSE INTERESTED.

HOPEFULLY THE PATCH WILL NOT ADVERSELY AFFECT THE REST OF SOLUSER9--I HAVEN'T NOTICED DEGRADATION TO SOLUSER9 BUT I DIDN'T USE SOLUSER9 MUCH BEFORE I INCORPORATED THE DELETE PATCH.

IT SHOULD BE NOTED THAT THE DELETE PATCH DESTROYS ONLY THE SECOND CHARACTER OF TWO-PART CHARACTERS AS CREATED ON THE SCREEN BY CONTROL CHARACTERS. THE BACKSLASH DELETE WAS NOT INCLUDED FOR THE SIMPLE REASON THAT I HAD NO PRESENT NEED FOR IT.

SINCERELY YOURS

Dick

DICK MOLLER
1371 W SIERRA
FRESNO, CA 93711

```
-LBB29,BB91
  BB29  CALL BB71        ;Continue CONIN below
  BB2C  JZ   BB29
  BB2F  CPI  80
  BB31  JNZ  BB36
  BB34  MVI  A,03
  BB36  CPI  8B
  BB38  JZ   BB49
  BB3B  CPI  8C          ;No change from SOLUSER9
  BB3D  JZ   C004
  BB40  ANI  7F
  BB42  CPI  7F
  BB44  RNZ
  BB45  STA  BB53
  BB48  RET
  BB49  PUSH B
  BB4A  MVI  B,0B
  BB4C  CALL C054
  BB4F  POP  B
  BB50  MVI  A,0D
  BB52  RET
  BB53  DCR  C
  BB54  DCR  C
  BB55  ??=  20
  BB56  PUSH B
  BB57  LDA  BB53
  BB5A  CPI  7F
  BB5C  JNZ  BB81        ;This patch replaces Dr. Byrams delete routine
  BB5F  LDA  BB54        ;This area is travelled twice for some
  BB62  CPI  7F
  BB64  JNZ  BB91        ;deletions.
  BB67  MVI  A,00
  BB69  STA  BB54
  BB6C  MVI  B,5F
  BB6E  JMP  BB8E
  BB71  CALL C02E        ;Call KSTAT
  BB74  STA  BB53        ;Deletes are intercepted and stored.
  BB77  STA  BB54        ;Watch out below_ adelete is coming through
  BB7A  RET              Return to continue CONIN
  BB7B  NOP
  BB7C  NOP
  BB7D  NOP
  BB7E  NOP
  BB7F  NOP
  BB80  NOP
  BB81  MOV  B,C
  BB82  LDA  BB55
  BB85  CPI  0D          ;No changes
  BB87  JNZ  BB8E
  BB8A  CMP  C
  BB8B  JZ   BB91
  BB8E  CALL C054
  BB91  POP  B
  BB92
```

Veit is High on Sol and Proteus

Stan Veit, who used to sell Sol computers in his Computer Mart store (now out of business), writes an interesting column in the Computer Shopper. In the December 15 issue, he mentions Proteus and his "beloved Sol." He says, "After my column on my beloved Sol computer, I received a letter from Stan Sokolow, informing me that PROTEUS, the Processor Technology Users Group was still very active. Naturally, I joined and am now receiving the Proteus/News, which is one of the best user group newsletters I have ever seen." He goes on to explain more about Proteus.

Thanks for the good words, Stan. Let me reciprocate by saying that Stan's column in the Computer Shopper is always interesting for me to read because he specifically talks about the computers we ol' timers remember, like the SWTPC 6800, the Altair, the IMSAI, the Digital Group, and so on. These names appear in the many ads in the Computer Shopper, but lots of newcomers don't have the foggiest idea of what they are. Stan is trying to give them a basis for understanding the used computer ads.

If any of you are bargain hunters, I can highly recommend the Computer Shopper. This not only has classified ads, it also has display ads from many small companies that often are selling good products at bargain prices. For example, the December issue shows a new 64K static RAM board similar to the Godbout RAM 17, for only $499 in kit ($40 more A&T). This uses the 6116 or 2016 type RAMs that are ultra-low power (0.5 amp per 64K).

Subscription is only $10 per year. Computer Shopper, P.O. Box F, Titusville, FL 32780.

WHO SAYS "CASSETTE TAPE IS DEAD"?

By Lewis Moseley

September 15, 1981

As soon as they get their disk systems, many hobbyists begin to look on their cassette machines as a sort of electronic red-headed stepchild. Disks are so much faster, the CP/M operating system so much more versitile; who needs the lowly tape?

Well, tapes can still be put to very good use, and there are a several programs in the Proteus Cassette Software Library to help you with this.

The DISKTAPE/TAPEDISK system was developed by Richard Greenlaw, a long-time regular contributor to PROTEUS. This system allows the writing of CP/M disk files onto cassette tapes, and the rereading of these tape files back into CP/M disk files.

The system has several advantages. First, you can have a cheap back-up method for your diskettes. A standard audio-grade C-90 cassette, which can be bought in quantity for about $1 each, will hold the contents of two 8" single-density disks, or about 500K.

Second, an interesting feature of the system is that all disk I/O is done through CP/M BDOS calls, and thus is device-independent. For those of you who don't speak fluent CP/M, this means that you can write a file from your 5" Northstar CP/M system onto a tape and give it to your buddy across town (or across the world; I have sent tapes as far as Africa) and he can load it onto his 8" CP/M system, and vice versa. This system has been tested with all of the following kinds of disk systems: 8" single and double density (several manu-facturers), Micropolis 5", Northstar 5", Vista 5", Helios 8" hard-sector. It is believed to work properly with all of these, and in fact, any CP/M system. The system requires a 20K or so CP/M system and a SOL or a compatable system with a CUTS board and CUTER.

The second program I want to discuss is called BYTE.ASM, and was written by me. It is designed to allow CP/M to read and write Proteus Standard Byte Access cassette files. Say that you have used ECBASIC for a couple of years and amassed a considerable quantity of software for it. Now you get your disk system. Do you have to throw away all of that software and start over? Nope, you just use ECBASIC to write the program out as a text file in byte access format (the ",T" option), and then use BYTE to read it into a CP/M file. Similarly, you might want to take a program you developed on your disk system and give it to a disk-less friend. Again, this is easy to do with BYTE. BYTE isn't a wonder program, though. It doesn't transLATE the program, it just transFERs it. So, you might still have a little syntactical syncronization to do (did I say that?).

I have mailed to PROTEUS a listing of the BYTE program, and it should be published with this letter. In addition, both BYTE and DISKTAPE/TAPEDISK are available on Proteus Library Cassette C-10 in CP/M format. As a matter of fact, C-10 is distributed in the Greenlaw DISKTAPE format, and I have sent out dozens without any problems of compatability. I'm sold on the programs, and if you will try them, you will be, too. The price for C-10 is $8.00 with an acceptable donation to the library and $18.00 without.

The Library still has the full CP/M Users Group disk library available on tape in the Greenlaw format. These cassettes are $10 each, with a catalog available for $6.

Please help me to help you by supporting the library with your contributions.

Regards,

Lewis Moseley Jr.

```
;*******************************************************
;**  This software available through:                **
;**    PROTEUS CASSETTE SOFTWARE LIBRARY              **
;**    C/O LEWIS MOSELEY, JR., LIBRARIAN              **
;**    2576 GLENDALE CT. NE                           **
;**    CONYERS, GEORGIA 30208                         **
;*******************************************************
;
;Librarian's note- This program is specifically intended  for
;transfer  of ASCII files between a CP/M system and a  system
;not equipped with CP/M.   Although it will work for transfer
;between  2  CP/M-equipped  systems,   the  DISKTAPE/TAPEDISK
;programs,  also  available through Proteus Cassette Software
;Library, would be much faster.
;
;   *** BYTE  ***
;
;PUBLIC DOMAIN SOFTWARE COURTESY OF:
;
;Lewis Moseley, Jr.
;2576 Glendale Ct. N.E.
;Conyers, Georgia 30208
;Version 2.0 of 11/22/79
;
;This  program  is a patch to allow CP/M's  PIP.COM  to  read
;SOL/CUTS byte access tape files into CP/M disk files, and to
;write  a  CP/M ascii text file into a SOL/CUTS  byte  access
;tape  file.  The routine uses the custom routine area within
;PIP,  located  from 103H-1FFH.   Note that most of  the  PIP
;options  can  be  used,  including  UC/LC  translation,  tab
;expansion,  echo to console,  line numbers, begin copying in
;mid-file,  stop copying in mid-file, etc.  Many applications
;programs, including most of Processor Tech's, do not  accept
;tabs,  and  so  be sure to use tab expansion when writing  a
;file for them.
;
;The  tape  format written,  and expected on  reads,  is  the
;Proteus Standard Byte Access format,  which ends text  lines
;with  a <CR> only.   Since CP/M ends lines with a <CR>-<LF>,
;this  tape driver supplies <LF>'s on reads,  and  suppresses
;them on writes.
;
;To write a tape, use the command:
;
;     A>PIP OUT:=<D:FN.FT>,EOF: <CR>
;
;         where  the name given is a valid  CP/M  unambiguous
;         file name.   The 'EOF:' should also be included, as
;         CP/M  does not always end-file properly without it,
;         and  this would cause part of the tape data  to  be
;         lost.
;
;Similarly, to read in a tape, use the command:
;
;     A>PIP <D:FN.FT>=INP: <CR>
;
;other standard PIP functions are also possible, such as:
;
;     A>PIP PRN:=INP: <CR>
;
;which would read a tape and print it, etc.
;
;At  the first attempt to access the tape,  you are  prompted
;for the tape file name on the console, and told to mount the
;tape.   PIP  checks  for  errors in disk  access,  and  this
;program checks for tape related errors.  Only brief messages
;are given, as there is only a limited space for this routine
;in PIP.
;
;To install the routine,  load it to disk and assemble it  as
;written.    Notice  that  it  just  barely  fits  the  space
```

CON'T FROM PAGE 9-WHO SAYS CASSETTE TAPE IS DEAD?

```
;available.  Use  STAT.COM to determine the size of PIP.COM,
;as this size varies according to the version of CP/M in use.
;Then,  use  DDT.COM to load it into the low part of PIP.COM,
;as follows:
;
;  A>DDT PIP.COM
;  -IBYTE.HEX              OPEN THE HEX FILE AS INPUT
;  -R                      READ HEX FILE IN WITH 0 OFFSET
;  <CTRL-C>                TO EXIT DDT
;  A>SAVE 28 NEWPIP.COM    SAVE AS NEW COMMAND FILE
;
;The  28  is the number of blocks to save for  PIP.COM  v1.4.
;Use the value you determined as directed above.
;
;This  program expects the normal CP/M entry point at  0005H,
;and also SOLOS or CUTER at 0C000H.  These assumptions can be
;changed by changing equates in the text file.
;
;
;
        ORG     103H            ;IN PIP CUSTOM AREA
;
;THESE ARE THE ENTRY POINTS FOR PIP'S INP: AND OUT:
        JMP     READ            ;READ FROM TAPE
        JMP     WRITE           ;WRITE TO TAPE
;
;ON ENTRY TO READ, DATA CONTAINS THE LAST BYTE
;READ IN FROM TAPE.  ON EXIT, IT CONTAINS THE
;CURRENT BYTE READ OR <CTRL-Z> ON ERROR OR EOF.
DATA    DB      0               ;DEFAULT VALUE
;
READ:   LDA     FCBAS           ;GET FILE ACCESS BYTE
        CPI     0FFH            ;FILE OPEN FOR READS?
        CNZ     OPEN            ;IF NOT, REOPEN
;HERE MEANS FILE IS OPEN FOR READ, DO IT.
        LDA     DATA            ;WAS LAST CHAR <CR>?
        CPI     CR
        JNZ     READ1           ;JUMP IF NOT
        MVI     A,LF            ;IS WAS, SO SUPPLY <LF>
        STA     DATA
        RET
READ1:  MVI     A,1             ;TAPE FILE 1
        CALL    RDBYT           ;GET NEXT BYTE
        JC      TERR            ;JUMP IF READ ERROR
        CPI     CTRLZ
        JZ      TERR1           ;JUMP IF EOF CHARACTER
;THE PROTEUS STANDARD TAPE FILE DOES NOT
;CONTAIN <LF>'S, BUT JUST IN CASE....
        CPI     LF
        JZ      READ1           ;IGNORE <LF>'S
        STA     DATA            ;ELSE NORNAL RETURN
        RET
;
;HERE ON EOF OR TAPE READ ERROR
TERR:   LXI     D,RDERR         ;ASSUME READ ERROR
        MVI     C,PRBUF         ;(M FLAG SET IFF E-O-F)
        CP      ENTRY           ;TELL USER IF READ ERROR
TERR1:  CALL    CLOSE           ;CLOSE FILE REGARDLESS
        MVI     A,CTRLZ         ;TELL PIP NO MORE DATA
        STA     DATA
        RET
;
WRITE:  MOV     A,C             ;CHECK FOR E-O-F CHAR
        CPI     CTRLZ
        JZ      CLOSE           ;JUMP IF TRUE
;CLOSE WHEN PIP SIGNALS EOF BY SENDING <CTRL-Z>,
;BUT NOT NECESSARY TO WRITE IT TO TAPE.
;ELSE MAKE SURE FILE IS PROPERLY OPEN
        PUSH    B               ;SAVE CHARACTER
        LDA     FCBAS           ;GET FILE ACCESS BYTE
        CPI     0FEH            ;FILE OPEN FOR WRITES?
```

```
        CNZ     OPEN            ;IF NOT, REOPEN
        POP     B               ;GET BACK CHAR
;HERE MEANS FILE IS OPEN AND READY FOR WRITE.
;FIRST, CHECK FOR SPECIAL CHARACTERS.
        MOV     A,C
        ANI     7FH             ;STRIP PARITY, IF ANY
        CPI     CR              ;ACCEPT <CR>'S
        JZ      WRIT1
        CPI     TAB             ;ACCEPT <TAB>'S, TOO
        JZ      WRIT1
        CPI     20H             ;IGNORE OTHER CONTROL CHARS
        RC
;HERE MEANS GOOD CHARACTER, WRITE IT
WRIT1:  MOV     B,C             ;SET UP CALL
        MVI     A,1
        CALL    WRBYT           ;DO IT!
        RET
;
;SUBROUTINES
;
;OPEN THE SPECIFIED FILE FOR READ OR WRITE
;(BUT NOT BOTH AT THE SAME TIME)
OPEN:   LXI     D,NAME          ;"ENTER FILE NAME" MSG
        CALL    REPLY           ;WRITE MSG & GET REPLY
        CALL    MFN             ;MOVE FILE NAME TO DHEAD
        LXI     D,RMSG          ;"READY TAPE..." MSG
        CALL    REPLY           ;WAIT FOR <CR>
        XRA     A               ;FORCE FILE CLOSED
        STA     FCBAS
        LXI     H,DHEAD         ;OPEN CUTS TAPE FILE
        MVI     A,1
        JMP     FOPEN           ;AND RETURN FROM THERE
;
;WRITE BUFFER FROM (DE), THEN GET REPLY
REPLY:  MVI     C,PRBUF         ;WRITE BUFFER CODE
        CALL    ENTRY
        LXI     D,BUFF          ;INPUT BUFFER
        MVI     C,RDBUF
        MVI     A,20            ;PLENTY OF SPACE
        STAX    D               ;FLAG BUFFER SIZE
        CALL    ENTRY           ;GET RESPONSE
        LXI     D,CRLFM         ;NOW DO <CR>,<LF>
        MVI     C,PRBUF
        JMP     ENTRY           ;AND RETURN FROM THERE
;
;CLOSE THE OPEN TAPE FILE
CLOSE:  MVI     A,1             ;CLOSE TAPE FILE #1
        JMP     FCLOS
;
;MOVE FILE NAME TO DHEAD
MFN:    LXI     H,BUFF+1                ;GET BUFFER LENGTH
        MOV     A,M
        ADD     L
        MOV     L,A             ;(HL)= LAST CHAR IN BUFFER
        INX     H               ;(HL)= JUST PAST LAST CHAR
        MVI     M,20H           ;FLAG END WITH SPACE
        LXI     H,BUFF+2                ;SKIP LENGTH BYTES
        LXI     D,DHEAD
        MVI     B,5
MFN1:   MOV     A,M             ;GET A CHAR
        CPI     21H             ;JUMP OUT ON FIRST SPACE...
        JC      MFN2            ;OR CONTROL CHAR
        STAX    D               ;ELSE STORE IT
        INX     D               ;BUMP POINTERS
        INX     H
        DCR     B               ;DONE 5 YET?
        JNZ     MFN1            ;JUMP IF NOT
        XRA     A               ;ELSE MARK END AND ABORT
        STAX    D
        RET
MFN2:   XRA     A               ;NAME MUST BE ZERO-FILLED
```

CON'T FROM PAGE 10-WHO SAYS CASSETTE TAPE IS DEAD?

```
             STAX    D
             INX     D
             DCR     B
             JNZ     MFN2
             INX     D                  ;PUT IN ZERO BYTE
             STAX    D
             RET
;
;***MESSAGES***
RDERR:  DB      CR
        DB      LF
        DB       'READ ERROR' ;FALL THRU
CRLFM:  DB      CR
        DB      LF
        DB      '$'
;
NAME:   DB       'Tape name:$'
;
RMSG:   DB       'Start tape, hit <CR>$'
;
SPACE   EQU     $        ;THE ADDRESS HERE MUST BE <= 200H


;***EQUATES***
;
ENTRY   EQU     5                  ;CP/M BDOS CALL
;
PRBUF   EQU     9                  ;FUNCTION CODES
RDBUF   EQU     10
;
SOLOS   EQU     0C000H                     ;SOLOS OR CUTER ROM ADDRESS
;
                                           ;SOLOS ROM ENTRY POINT ADDRESSES
FOPEN   EQU     SOLOS+07H                  ;BYTE ACC. FILE OPEN .
FCLOS   EQU     SOLOS+0AH                  ;FILE CLOSE
RDBYT   EQU     SOLOS+0DH                  ;READ A BYTE
WRBYT   EQU     SOLOS+10H                  ;WRITE A BYTE
;
                                           ;SOLOS SYSTEM RAM LOCATIONS
DHEAD   EQU     SOLOS+82CH                 ;DUMMY TAPE HEADER
FCBAS   EQU     SOLOS+855H                 ;TAPE FILE CONTROL CHAR
BUFF    EQU     SOLOS+963H                 ;PUT IN TAPE BUFFER #2
;
TAB     EQU     09H
LF      EQU     0AH
CR      EQU     0DH
CTRLZ   EQU     1AH
```

Sol Vanishing Trick
It Almost Works

In the last issue, I reported an easy modification to make the Sol into a 64K machine by making the Sol's dedicated address space go away. This trick that disables the Sol's address decoder apparently has some side effects that may or may not interfere with the way you want to use it. One reader reports that with the mod installed, Sol not only responds to the C000 addresses it normally does, but also to the D's and E's blocks. If you don't refer to these addresses except when the Sol is "vanished" out, this may not be a problem, but it certainly plays havoc with the system if you have the Sol operating in supposedly normal mode.

Another problem that I suspect, but haven't investigated personally, is that with the Sol in "normal" mode, writing into the screen RAM will also write into the overlapping S-100 bus RAM. With the Sol vanished, writing into the bus RAM should have no effect on the screen, though.

The Micro Complex 80/64 modification board, described in the cover story of this issue, is a more sophisticated way of getting Sol's address space out of the way, as well as making many other improvements to the Sol. Look into it.

SOLUTION TO THE FILE DRIVER FOR NEC SPINWRITER
M.K. Gauthier

Dear Stan,

I found a solution to the file driver problem for my NEC Spinwriter.

After talking with you about two weeks ago about the problem, I went to work trying to make the NEC file driven on the H-5 disk work with "Mail Master" and "Mail Sort". I guess I am not smart enough to get it to work. So I went back to the Sol3 (high speed) file driver on "Mail Master" and "Mail Sort". By placing switch 5 on the 8 switch DIP switch inside the NEC to the "down" position, the printer works fine, except it only prints left to right. For "WordWizard" and my custom software I place the switch to the "up" position and use the H-5 file driver and the printer prints bi-directionally. To get to the switch easily, I made a simple tool by unbending a paper clip and placing a small hook on the end. It just slides down the front of the printer making easy contact with the DIP switches.

I have not tried this yet with the "General Ledger" program but I would expect the same results as with the "Mail Master" and "Mail Sort" programs.

I hope this information is helpful to others in the users group.

Question: Is anyone working on an Electronic Dictionary to be used with "Wordwizard"? I would be interested in purchasing it.

Keep up the good work Stan,

GAUTHIER INDUSTRIES

Dr. Michael K. Gauthier, P.E.
President

MKG/ww

9550 Gallatin Road
Downey, California 90240
213-923-0131

CORRECTION ON EPSON MX-80:

Wordcraft

534-2212

Please note the error in the pin connection chart for the Epson MX-80 (Proteus News, Jan.-Feb. '81, IV:1). The correct chart is the following:

SOL Parallel Port

1	2	16	17	18	19	20	21	22	23	24	25
↯	↯	↯	↯	↯	↯	↯	↯	↯	↯	↯	↯
17	23	11	1	9	8	7	6	5	4	3	2

MX-80 Parallel Port

PRINT SPOOLING FOR N*
Wm. David Armstrong

Armstrong Corporation

125-F Melody Lane (P.O. Box 533), Costa Mesa, CA 92627 - (714) 752-8050

September 25, 1981

Dear Stan;

Included with this letter is an assembly language I/O routine to perform print spooling when used with a standard North Star DOS in a SOL system which has memory in the 0000H to 1FFFH area. As an added feature, the I/O routines include a RENAME function for those using North Star's release 4 DOS. The rename function permits file names in the disk directory to be changed with the following simple command:

```
     RN (OLDNAME,unit) (NEWNAME)<cr>
```

Ben Stapleton's program (PROTEUS vol.4, #3/4) for obtaining a hard copy of the SOL's video screen reminded me of a similar program that I wrote some time back. I have included it with this letter for the benefit of North Star and CP/M users who may find it a little more convenient to use (though not quite as compact as Ben's version). During any program that uses DOS I/O (BASIC, etc.), provided that a printer is on line (Sol's LOCAL light off, etc.), simply pressing the LOAD key will cause a copy of the current video screen contents to be dumped to the printer.

In reference to that same issue of PROTEUS, I must add an addendum to the LOAN CALCULATION PROGRAM which I submitted. There are three statements which should be modified. First, add the following line: 185 LET M=0. This will zero the value of "months" after a calculation involving loan terms of partial years. Secondly, change the first statement on line 390 from I=.1 to I=.15 in order to permit the program to deal with today's 20%+ interest rates. Finally, delete the FILL statement on line 490. This statement was intended for use with North Star release 4 BASIC only. It's function is to zero the line counter after a direct cursor manipulation. The same effect can probably be had by setting the console line length to its maximum value. This can be done in North Star BASIC by adding the following line:
35 LINE 130.

I would also like to add a post script to Deslar Patten's letter in that same issue of PROTEUS concerning the "clear to end of line" on carriage return problem in the SOL. He proposes three solutions. To these, I would add a fourth: Send an ESCAPE-1-0 sequence to the video screen instead of a carriage return. This is the SOLOS direct cursor addressing method of returning to the beginning of a line. Many programs, such as Peter Roizen's excellent TMAKER II program (available from Lifeboat Associates) allow the user to specify multiple character sequences in lieu of standard control characters. In his program, for example, one would specify ESC (1BH), followed by CTRL-A (01H), followed by one null, to effect a carriage return instead of CTRL-M (0DH).

Concerning the SOLOS direct cursor addressing capability, I found that some readers were surprised to see it used in my LOAN CALCULATION PROGRAM. Here are a few additional uses of the SOLOS VDM driver: (reference: SOLOS User Manual page 5.4)

Function to print string in REVERSE VIDEO.

```
        10 REM EXAMPLE:
        20 A$="HELLO" \ PRINT FNR$(A$)
        30 END
        9000 DEF FNR$(X$)
        9005 FOR I=1 TO LEN(X$) \ X=ASC(X$(I,I))+128
        9010 PRINT CHR$(27)+CHR$(5)+CHR$(X),
        9015 NEXT \ RETURN "" \ FNEND
```

Function to set VIDEO DISPLAY SPEED.

```
        10 REM EXAMPLE:
        20 X=FNS(10) \ PRINT "HELLO" \ X=FNS(0)
        30 END
        9000 DEF FNS(X)
        9005 IF X<0 OR X>255 THEN X=0
        9010 PRINT CHR$(27)+CHR$(8)+CHR$(X),
        9015 RETURN 0 \ FNEND
```

Function to print LITERAL VALUE OF CONTROL CHARACTERS.

```
        10 REM EXAMPLE:
        20 PRINT FNL$(13)
        30 END
        9000 DEF FNL$(X)=CHR$(27)+CHR$(6)+CHR$(X)
```

Function to effect DIRECT CURSOR ADDRESSING.
If Y (line) value exceeds 15, then remain on current line.

```
        10 REM EXAMPLE:
        20 PRINT FNA$(25,6),"HELLO"
        30 END
        9000 DEF FNA$(X,Y)
        9005 IF Y<16 THEN 9015 \ X$=CHR$(27)+CHR$(1)+CHR$(X)
        9010 PRINT X$, \ Y=CALL(49462) \ X$="" \ GOTO 9020
        9015 X$=CHR$(27)+CHR$(1)+CHR$(X)+CHR$(27)+CHR$(2)+CHR$(Y)
        9020 RETURN X$ \ FNEND
```

NOTE: The call statement at line 9010 uses SOLOS routines to remove the cursor at the same time the cursor position is moved to the beginning of the line. This is useful in some programs, and unnecessary in others. The CALL statement can be deleted if not needed.

NOTE: This function may not work properly if the scroll counter has changed since the last Clear Screen operation, [Clear Screen: PRINT CHR$(11)] Additional statements in the direct cursor addressing function could take into account the value of the scroll offset (BOT; 0C80AH).

As a final item, many individuals have asked for my advice as to what Assembler is best to use with the SOL-20. As far as I'm concerned, you can do no better than ALS-8 by Processor Technology. Micro Complex (25651 Minos St., Mission Viejo, CA 92691) can supply you with a version relocated to run at 00H, using SOL cassette or North Star DOS (any version). I have heard many users compare the VDM file editor in ALS-8 with the popular and powerful Electric Pencil word processor editor.

Sincerely,

Wm. David Armstrong

Wm. David Armstrong
encl.

```
2900                0005 * North Star release 4 Disk Operating System
2900                0010 * I/O routines with SPOOLER for print output
2900                0015 *
2900                0020 * (C)1978 ARMSTRONG CORPORATION
2900                0025 *         P.O. Box 10533
2900                0030 *         Costa Mesa, CA 92627
2900                0035 *
2900                0040 * In N.S. rels 4 DOS versions ONLY, the following routines
2900                0045 * labeled DEL and RNAME allow for the addition of a new
2900                0050 * and useful system command which permits the renaming of
2900                0055 * disk files.  To activate the use of these routines, the
2900                0060 * following patches will have to be made:
2900                0065 *
2900                0070 *    1. Change the "DE" jump in the DOS command table at
2900                0075 *       210FH from 2141H to the address of DEL (2900H).
2900                0080 *    2. Increase the value at 20B7H to reflect the new
2900                0085 *       number of command table entries (from 0EH to 0FH).
```

CON'T FROM PAGE 12-PRINT SPOOLING FOR N*

```
2900           *    3. Add the new command table entry at 2141H by
2900           *       entering the ASCII values of the command name,
2900           *       i.e., "RN", followed by the jump address to RNAME
2900           *       at 290CH.  NOTE: There is also room for two more
2900           *       command table entries if you wish to add them.
2900           *       You may want to consider adding "BA" for a jump
2900           *       to BASIC at 2A04H, and "BC" for a jump to 2A00H.
2900           *       If additions are made, don't forget to repeat
2900           *       step 2, above. (Also, don't forget that all
2900           *       addresses are entered into memory low byte first.)
2900           *
2900           * NOTE: Do not make these changes unless your DOS is rel.4
2900           *
2900 CD A5 23  DEL    CALL   23A5H
2903 CD 4D 27         CALL   274DH
2906 CD FA 23         CALL   23FAH
2909 C3 04 24         JMP    2404H
290C           *
290C CD A5 23  RNAME  CALL   23A5H  RENAME EXTENSION
290F E5               PUSH   H
2910 3A F0 28         LDA    28F0H  UNIT#
2913 F5               PUSH   6
2914 CD B7 23         CALL   23B7H
2917 F1               POP    6
2918 32 F0 28         STA    28F0H
291B C3 BF 22         JMP    22BFH
291E           *
291E           * The SPOOLER employs the memory from 0000H to 1FFFH as
291E           * a circular buffer for output to a printer.  Therefore,
291E           * it is assumed that memory is available in this area and
291E           * that there is no conflicting memory use.
291E           *
291E           * The DOS JUMP TABLE from 200DH to 2019H will, of course,
291E           * have to be patched to reflect the new routine addresses:
291E           *        200D   JMP   OUTPB
291E           *        2010   JMP   KEYIN
291E           *        2013   JMP   TINT
291E           *        2016   JMP   CTRLC
291E           *
291E C3 D5 C0  TINT   JMP    0C0D5H  ERASE SCREEN (For std. SOLOS only.)
2921           *
2921 00 00     INPTR  DW     00H    SPOOLER INPUT POINTER
2923 00 00     OTPTR  DW     00H    SPOOLER OUTPUT POINTER
2925 00        WAITF  DB     00H    PRINTER BUSY FLAG


2926 00        INPF   DB     00H    INPUT FLAG
2927 00        OUTF   DB     00H    OUTPUT FLAG
2928           *
2928 E6 03     KEYIN  ANI    03H
292A 32 26 29         STA    INPF
292D CD A2 29  WAITI  CALL   SPOUT
2930 3A 26 29         LDA    INPF
2933 CD 22 C0         CALL   AINP   0C022H  SOLOS AINP RTN.
2936 CA 2D 29         JZ     WAITI
2939 FE 80            CPI    80H
293B CA 04 C0         JZ     0C004H  OPTIONAL JUMP TO SOLOS ON MODE SELECT
293E FE 03            CPI    03H    CONTROL-C?
2940 CC 6A 29         CZ     STOP   KILL PRINTER OPERATION
2943 E6 7F     CNT1   ANI    7FH
2945 FE 7F            CPI    7FH    61H FOR UPPER CASE ONLY
2947 D8               RC
2948 D6 20            SUI    20H
294A C9               RET
294B           *
294B E6 03     OUTPB  ANI    03H
294D C2 55 29         JNZ    CNT2
2950 CD 19 C0         CALL   0C019H  SOLOS SOUT RTN.
2953 78               MOV    A,B
2954 C9               RET
```

```
2955 FE 01     CNT2   CPI    01H    CHECK FOR PRINTER PSEUDO PORT
2957 CA 73 29         JZ     SPINP  SPOOL IT IF PRINTER PORT SELECTED
295A CD 1C C0         CALL   AOUT   OTHERWISE, USE 0C01CH SOLOS AOUT RTN.
295D 78               MOV    A,B
295E C9               RET
295F           *
295F CD A2 29  CTRLC  CALL   SPOUT
2962 CD 1F C0         CALL   SINP   0C01FH SOLOS SINP RTN.
2965 E6 7F            ANI    7FH
2967 FE 03            CPI    03H
2969 C0               RNZ
296A E5        STOP   PUSH   H
296B 2A 21 29         LHLD   INPTR
296E 22 23 29         SHLD   OTPTR
2971 E1               POP    H
2972 C9               RET
2973           *
2973 C5        SPINP  PUSH   B      SPOOLER BUFFER INPUT
2974 D5               PUSH   D
2975 E5               PUSH   H
2976 2A 21 29         LHLD   INPTR
2979 70               MOV    M,B
297A CD E8 29         CALL   INCPR
297D 22 21 29         SHLD   INPTR
2980 CD E8 29         CALL   INCPR
2983 EB               XCHG
2984 2A 23 29         LHLD   OTPTR
2987 CD E9 29         CALL   INCPR+1
298A C2 D6 29         JNZ    EXIT
298D CD DB 29  WAIT2  CALL   PBUSY
2990 C2 8D 29         JNZ    WAIT2
2993 AF               XRA    A
2994 32 25 29         STA    WAITF
2997 4F               MOV    C,A
2998 CD A2 29  OT256  CALL   SPOUT
299B 0C               INR    C
299C C2 98 29         JNZ    OT256
299F C3 D6 29         JMP    EXIT
29A2           *


29A2 CD DB 29  SPOUT  CALL   PBUSY  SPOOLER BUFFER OUTPUT
29A5 C0               RNZ
29A6 AF               XRA    A
29A7 32 25 29         STA    WAITF
29AA C5               PUSH   B
29AB D5               PUSH   D
29AC E5               PUSH   H
29AD 2A 21 29         LHLD   INPTR
29B0 EB               XCHG
29B1 2A 23 29         LHLD   OTPTR
29B4 CD E9 29         CALL   INCPR+1
29B7 CA D6 29         JZ     EXIT
29BA 46               MOV    B,M
29BB 3E 01            MVI    A,01H  SELECT OUTPUT DEVICE
29BD CD 1C C0         CALL   AOUT   0C01CH SOLOS AOUT ROUTINE
29C0 CD E8 29         CALL   INCPR
29C3 22 23 29         SHLD   OTPTR
29C6 78               MOV    A,B
29C7 FE 0D            CPI    0DH
29C9 C2 D6 29         JNZ    EXIT   This instruction is for ETX/ACK protocol
29CC           *             printers (QUME, DIABLO, NEC, etc.) only.
29CC           *             The instruction should be "JMP EXIT" for
29CC           *             printers which employ CLEAR-TO-SEND or
29CC           *             DATA-SET-READY protocols (TI-810, ANADEX
29CC           *             PRINTERM, BASE II, etc.).
29CC           *
29CC 3E 01            MVI    A,01H  SELECT OUTPUT PORT
29CE 32 25 29         STA    WAITF
29D1 06 03            MVI    B,03H
```

CON'T FROM PAGE 13-PRINT SPOOLING FOR N*

```
29D3 CD 1C C0        0770        CALL   AOUT   (0C01CH) SEND ETX TO PRINTER
29D6 E1              0775 EXIT   POP    H
29D7 D1              0780        POP    D
29D8 C1              0785        POP    B
29D9 78              0790        MOV    A,B
29DA C9              0795        RET
29DB                 0800 *
29DB                 0805 * The routine following is for ETX/ACK printers.  If the
29DB                 0810 * printer you have uses DATA-SET-READY or CLEAR-TO-SEND,
29DB                 0815 * substitute the routine below in place of the other:
29DB                 0820 *
29DB                 0825 * PBUSY  IN   0FBH  (Get serial status)
29DB                 0830 *        ANI   02H  (20H for CLEAR-TO-SEND printers)
29DB                 0835 *        RET
29DB                 0840 *
29DB 3A 25 29        0845 PBUSY  LDA    WAITF  PRINTER BUSY?
29DE B7              0850        ORA    A
29DF C8              0855        RZ
29E0 3E 01           0860        MVI    A,01H  SELECT INPUT DEVICE
29E2 CD 22 C0        0865        CALL   AINP   0C022H SOLOS AINP RTN.
29E5 FE 06           0870        CPI    06H    ACK RECEIVED?
29E7 C9              0875        RET
29E8                 0880 *
29E8 23              0885 INCPR  INX    H      INCREMENT POINTER,
29E9 7C              0890        MOV    A,H    AND COMPARE HL & DE
29EA E6 1F           0895        ANI    1FH    MODULUS 2000
29EC 67              0900        MOV    H,A
29ED BA              0905        CMP    D
29EE C0              0910        RNZ
29EF 7D              0915        MOV    A,L
29F0 BB              0920        CMP    E
29F1 C9              0925        RET
29F2                 0930 *
29F2                 0935 * END OF PROGRAM *
```

```
2900                 0005 * PRINT VIDEO SCREEN ROUTINE FOR SOL-20
2900                 0010 * (Using North Star or CP/M DOS systems)
2900                 0015 *
2900                 0020 * (C)1980 ARMSTRONG CORPORATION
2900                 0025 *          P.O. Box 10533
2900                 0030 *          Costa Mesa, CA 92627
2900                 0035 *
2900                 0040 * To integrate the print screen routine into your DOS,
2900                 0045 * provide a jump to SCNPT in an appropriate place in
2900                 0050 * your Console input routines.  For example:
2900                 0055 *
2900 E6 03           0060 INPUT  ANI    03     This is the North Star version.
2902 32 13 29        0065        STA    FLAG   Save input device number.
2905 3A 13 29        0070 INP1   LDA    FLAG
2908 CD 22 C0        0075        CALL   AINP   SOLOS AINP ROUTINE (0C022H)
290B CA 05 29        0080        JZ     INP1   Loop until character received.
290E FE 8C           0085        CPI    140    SOL-20 LOAD KEY PRESSED?
2910 CA 14 29        0090        JZ     SCNPT  If yes, then jump to screen print routine.
2913                 0095 * (Regular program flow continues here)
2913                 0100 *
2913 00              0105 FLAG   DB     0
2914                 0106 CNTLC  EQU    $      FOR DEMO PURPOSES ONLY
2914                 0107 *
2914                 0110 VDADD  EQU    0C11CH
2914                 0115 CREM   EQU    0C136H
2914                 0120 OUTPB  EQU    0200DH
2914                 0125 NCHAR  EQU    0C808H
2914                 0130 BOT    EQU    0C80AH
2914                 0135 *
2914 E5              0140 SCNPT  PUSH   H      SAVE MOST REGISTERS
2915 D5              0145        PUSH   D
2916 C5              0150        PUSH   B
2917 CD 36 C1        0155        CALL   CREM   REMOVE CURSOR
291A 3A 0A C8        0160        LDA    BOT    A HAS SCROLL OFFSET
291D 3C              0165        INR    A
291E 21 C0 CB        0170        LXI    H,0CBC0H  HL HAS NORMAL START OF SCREEN
2921 11 40 00        0175        LXI    D,64   DE HAS ONE LINE DIFFERENCE
2924 19              0180 SCN1   DAD    D      ADD ONE LINE
2925 3D              0185        DCR    A      FOR EACH LINE OFFSET
2926 C2 24 29        0190        JNZ    SCN1   LOOP UNTIL DONE
2929 4F              0195        MOV    C,A    CLEAR REG. C
292A 11 00 04        0200        LXI    D,1024  DE GETS SCREEN COUNTER
292D 79              0205 SCN2   MOV    A,C    C KEEPS #-CHARACTERS PER LINE
292E E6 3F           0210        ANI    63     NO MORE THAN 64 FOR SOL-20
2930 4F              0215        MOV    C,A    SAVE IT
2931 C2 42 29        0220        JNZ    SCN3   ONWARD IF NOT 64
2934 06 0D           0225        MVI    B,13   CARRIAGE RETURN
2936 3E 01           0230        MVI    A,1    FOR PRINTER
2938 CD 0D 20        0235        CALL   OUTPB  GOES OUT TO PRINTER
293B 06 0A           0240        MVI    B,10   FOLLOWED BY LINE FEED
293D 3E 01           0245        MVI    A,1
293F CD 0D 20        0250        CALL   OUTPB
2942 7E              0255 SCN3   MOV    A,M    NOW REGULAR BUSINESS
2943 E6 7F           0260        ANI    7FH    GET CHARACTER & STRIP PARITY
2945 FE 20           0265        CPI    32     TEST FOR CONTROL CHARACTER
2947 D2 4C 29        0270        JNC    SCN4   ONWARD IF OKAY
294A 3E 3F           0275        MVI    A,63   OTHERWISE IT'S A "?"
294C 47              0280 SCN4   MOV    B,A    SAVE IT IN B
294D 3E 01           0285        MVI    A,1    SET FOR PRINTER OUTPUT
294F CD 0D 20        0290        CALL   OUTPB  AND SEND IT
2952 CD 14 29        0295        CALL   CNTLC  CHECK FOR CONTROL-C
2955 CA 68 29        0300        JZ     SCN6   ONWARD IF SO
2958 23              0305        INX    H      POINT TO NEXT CHARACTER
2959 0C              0310        INR    C      AND BUMP LINE COUNTER
295A 7C              0315        MOV    A,H    CHECK FOR WRAP AROUND
295B FE D0           0320        CPI    0D0H   ARE WE THERE?
295D C2 62 29        0325        JNZ    SCN5   ONWARD IF NOT
2960 26 CC           0330        MVI    H,0CCH  OTHERWISE, WRAP AROUND
2962 1B              0335 SCN5   DCX    D      ONE LESS CHARACTER TO WORRY US
2963 7A              0340        MOV    A,D    SINCE DCX SETS NO FLAGS,
2964 B3              0345        ORA    E      THIS IS HOW WE CHECK FOR ZERO IN DE
2965 C2 2D 29        0350        JNZ    SCN2   MORE WORK AHEAD
2968 CD 1C C1        0355 SCN6   CALL   VDADD  PUT CURSOR BACK
296B 7E              0360        MOV    A,M
296C F6 80           0365        ORI    80H
296E 77              0370        MOV    M,A
296F C1              0375        POP    B      RESTORE ALL REGISTERS
2970 D1              0380        POP    D
2971 E1              0385        POP    H
2972 C3 00 29        0390        JMP    INPUT  PRETEND WE'RE STILL WAITING FOR KEYINP
2975                 0395 *
2975                 0400 * NOTE: For CP/M systems, lines 0235, 0250, 0290 and
2975                 0401 * 0295, above, will have to be modified to reflect
2975                 0405 * the differences in the output routines of the two
2975                 0410 * operating systems.  One method might look like this:
2975                 0411 *
2975                 0415 * OUTPB   PUSH   H
2975                 0420 *         PUSH   D
2975                 0425 *         PUSH   B
2975                 0430 *         MOV    C,B
2975                 0435 *         CALL   LIST
2975                 0440 *         POP    B
2975                 0445 *         POP    D
2975                 0450 *         POP    H
2975                 0455 *
2975                 0460 * END OF PROGRAM *
```

SOL KEYBOARD TONE CIRCUIT / WITH 4 TONE LEVELS
by H. Leon Winter

Dear Stan, Nov 22, '81

I've been enjoying Proteus / News for a long time. Being a fellow Sol owner, I'm most thankful that this group exists! Except in maters of dues and membership, I've not written in before as I live in the Philippines and am sort of out of alot that is going on. As it happened though, I was discussing a couple of points in the latest issue (March/April '81) that came to me receintly with my wife and she encouraged me to put them in a letter to you so here it is.

First a little about my particular system. It's a Sol-20 with 36K of RAM in a Heintz verity of boards. I use duel cassettes (Superscopes) as my mass storage and a Ledex video 100 for display. Printed output comes through a simple interface connected to Sol's parallel port and to a 20 year old ITEL (Selectric). The old printer accounts for the random unstable printing you are looking at; not the Sol. I hope to pick up a used Hytype I in about a year which will connect nicely to a duel parallel port S-100 board that I have.

I was most encouraged by the mention of the coming 24 X 80 video modification for Sol mentioned in the Mar/Apr issue. I say "Hurray" for the Calif mfr who is investing in this. I believe he will immediatly be sold out!

I have a simple fix for the dead key problem the Emile Roth asked about. In the 3 years that I've had my Sol, I've had about 5 diferent keys do this. What I've done is to remove the metal keyboard support frame (about 16 little screws) from the board. Then cut a small disk from household aluminum foil the right size and glue to the aluminum already on the key bottom. I use ordinary household glue with never a repeat of the problem in the same place. Be sure the shinny side of the new pad it toward the circuit board. Also while you have the keys separate from the board; this is a good time to do cleaning of the whole card. Keyboards get dusty you know!

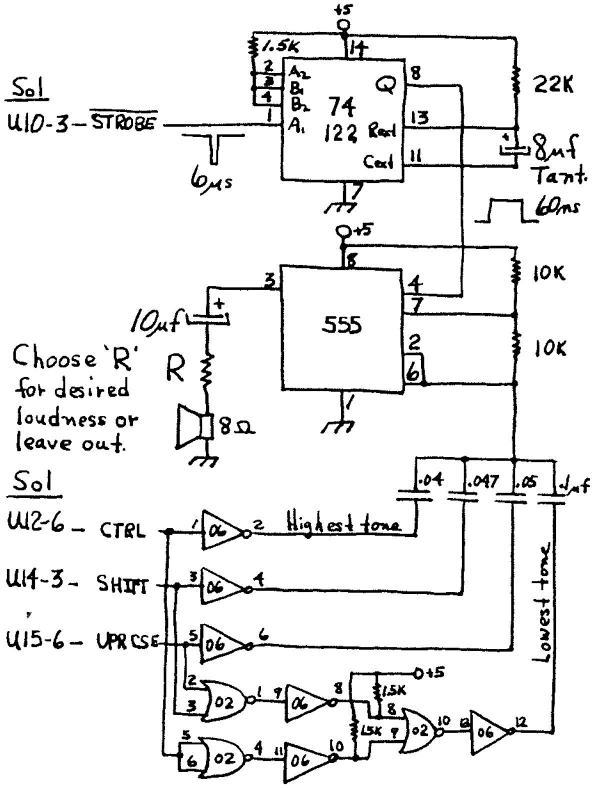
Adding tones to Sol's keyboard is not a new trick. I've seen many articles on it, but included is a schematic for one I did that may be of interest to some as it gives different tone levels to the 4 main keyboard outputs. These are unshifted, shifted (shift lock), alpha shift (upper case) and control. These were simply put on a small board and attached to the right underside of the keyboard along with a very small speaker. There is plenty of sound so the speaker does not need to be put outside. I soldered directly to points on the underside of the board. This has been working trouble free now for over 2 years.

In looking at "Bells on my Sol" (Mar/Apr), I think the idea of a bell is good. What bothers me is tying up the parallel port for it. Has anyone considered using one of the 2 unused outputs from the output port decoder shown on X-15? This is U34 (74LS138) pins 7 and 11. In fact, on my drawing, pin 11 says in parenthesis (alarm). I've not checked it out, but I think that pin 7 would decode to port out FF and pin 11 would be port out FC. This would be a neat way to get a bell, etc, and not give up the parallel port! Something for nothing? I'd like to hear from someone who may have done this.

My thanks to Albert Woodhull for the fine program to renumber lines in Software #1. Now if we had just one more fix, SOFT1 would be really convient to use. This would be for the editor to auto tab to the next field as source is being typed in.

Thank you, Stan, for continueing to publish a fine source of information, reviews and how to's. It helps make owning a Sol special.

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



IC's
74122
7406
7402
555

Sol Keyboard Tone Circuit
/w 4 tone levels
Refer to X-22
H Leon Winter, Jan '80

LATE POST-SCRIPT TO 24x80 REVIEW
Continued from page 3

I just spent two days working with the 24x80 module after getting it back from Micro Complex. The bug I had did turn out to be a subtle timing problem that was fixed in later Sols but not on my old one, according to Bob Hogg. He will report the improvement in Proteus. I have made a quicky patch to let PTDOS talk with the 24x80 screen, and have no problems. Programs like EDIT with their own video driver won't work until they are changed, but the character-oriented console I/O works like a charm. It even beeps instead of giving you the little bell symbol on the screen. More in the future issues.

PTDOS TO CP/M FILE TRANSFER
by Don L. Finley

Enclosed is a check for another year's subscription to Proteus.

I noted with interest the article by Mr. Zeratsky in the Nov./Dec. 198Ø Proteus dealing with converting PTDOS text files to CP/M files. I encountered the same problem several months ago when I first started playing with CP/M. However, I took a somewhat different approach. Enclosed is the listing of a program (DOWNLOAD) I wrote to ease the burden somewhat.

The program resides on a CP/M disk configured for 3ØK memory. By restricting CP/M to low memory it is possible to have both CP/M and PTDOS active at the same time.

To use the program I first BOOT on a PTDOS disk, remove the disk, UPPER CASE-REPEAT and BOOT on the 3ØK CP/M disk. I then insert the PTDOS disk in drive 1 and type DOWNLOAD CPMFILE PTDOSFIL. The rest is automatic. As the program is being downloaded (note the implied relationship between PTDOS and CP/M) the program removes excess spaces from the file, replacing them with tabs and appends the necessary line-feeds upon encountering carriage-returns.

In another vein, I have found SOLOS very useful in transporting CP/M files from one SOL system to another, especially when there is a disk incompatability between the two systems. To write a file onto a SOLOS tape, I use DDT to load the file into memory, then execute a GCØØ4 to get to SOLOS, set the tape file type, and SAVE the file on the tape. An example might be as follows:

```
A>DDT EDIT.COM            (Use DDT to load the file into mem.)
DDT Version whatever.     (DDT commercial)
NEXT  PC                  (DDT now prints the)
13FF Ø1ØØ                 (end address, start address)
-GCØØ4                    (go to SOLOS)
>SET TY=43                (in SOLOS, set file type to C)
>SAVE EDIT 1ØØ 13FF       (save on tape from 1ØØ to 13FF)
>EX Ø                     (return to CP/M)
```

Going from SOLOS to CP/M might look as follows:

```
A>UPPER-CASE REPEAT       (leave CP/M go to SOLOS)
                          (screen  is  cleared)
>GET                      (get the first file)
EDIT C Ø1ØØ 12FF          (SOLOS reads the file in)
>EX Ø                     (return to CP/M)
A>SAVE 19 EDIT.COM        (save 19 256 byte blocks as EDIT.COM)
                          (note 12FF = 13 hex 256 bytes blocks)
                          (and  13 hex = 19 decimal)
```

And that's all there is to it. CP/M Text files as well as executable files can be transported in this manner.

Now for a commercial message. After having been exposed to PT's fine editor, EDIT, I found it very frustrating trying to use CP/M's #%&#! editor. Therefore, one of my first CP/M acts was to write an EDIT-like editor for CP/M. It has most of the regular EDIT features such as full cursor control with forward and backward scrolling, etc. In addition, it maintains tabs internally as tabs while expanding them on the screen. This can significantly reduce the amount of disk space required for a text file. Additional features include tag defined blocks, block moves, block deletes, block listing to a printer, writing a block to a side disk file, inserting a side file, string searching forward and backwards plus several other goodies.

The editor is presently configured for use with CP/M on the SOL. I also have a version for a non-SOL (hiss) system using a TELEVIDEO terminal. Upon request I can probably configure it for other terminals which have an addressable cursor. If anyone is interested, I am selling the editor for $34.95 on a HELIOS compatible disk. The editor can also be obtained in an 8-inch soft sector format, single or double density, diskette from Micro-Products and Systems, 23Ø7 Center St. Kingsport,Tn. 3766Ø. Or if they are willing to accept a slight delay I can furnish the editor on N* disks. Or I can send it on a SOLOS-CUTS tape as outlined above. I will also include several other utilities with the editor.

Another program which may interest someone is a disk recovery program I wrote for CP/M. I have already used it several times to unerase erased files and in one instance to recover a valuable text file from a crashed disk. At the present time, it is still in the development stage - i.e. I'm still adding goodies to it. As soon as I "complete" the program (named LAZARUS) I will offer it for sale. In the mean time, if anyone needs a disk recovered or some un-ERAsing done, they should send the disk to me, describe the nature of the problem and I'll see what I can do. Note, once the crash or erasure has occured don't write anything else to the disk before it has been recovered. All files on the disk will be kept in strict confidence. Not only that, but I won't charge much, if anything.

Hang in there and keep up the good work.

Sincerely Yours,
Don L. Finley
31Ø Willow St.
Mt. Carmel, TN.
37642
(615) 357-3355

```
;*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+
;                  PTDOS TO CPM FILE TRANSFER
;  Don L. Finley   1Ø-22-8Ø
;
; THE FOLLOWING PROGRAM WILL DOWNLOAD A TEXT FILE
; FROM PTDOS TO CPM. THE PROGRAM WILL AUTOMATICALLY ADD
; LINE-FEEDS AFTER EACH CARRIAGE-RETURN. IN ADDITION IT WILL
; REMOVE UNNECESSARY SPACES AND COMPRESS THEM INTO TABS.
; THE FOLLOWING RULES APPLY TO SPACE COMPRESSION:
;   SPACES WILL BE REMOVED AND A TAB INSERTED
;   PROVIDED TWO CONSECUTIVE SPACES ARE FOUND
;   UNLESS PRECEEDED BY A SEMI-COLON OR A '
; NOTE ALSO THAT A GROUP OF SPACES WILL BE REPLACED WITH ONLY
; A SINGLE TAB.
;
; TO RUN THE PROGRAM:
; 1. BOOT ON PTDOS DISK IN DRIVE Ø
; 2. UPPERCASE-REPEAT
; 3. REMOVE DISK AND REPLACE WITH A 3ØK CPM SYSTEM DISK
; 4. BOOT ON 3ØK CPM IN DRIVE Ø
; 5. PLACE PTDOS DISK IN DRIVE 1
; 5. TYPE DOWNLOAD CPMNAME.EXT PTNAME
; 6. SET BACK AND WAIT TILL FINISHED
;*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+*+
        ORG 1ØØH
        LXI SP,STACK
        LXI D,MSG1          ;PRINT SIGN-ON MESSAGE
        MVI C,PSTRIN
        CALL BDOS
        CALL ASKOK          ;SEE IF USER WANTS TO CONTINUE
        JNZ Ø               ;IF NOT, THEN A RETURN TO CPM
        LDA 8ØH             ;GET NUMBER OF CHRS TYPED ON K'BORD
        ORA A               ;SEE IF ANYTHING ENTERED
        JZ ERR1             ;NOTHING ENTERED- BOMB OUT
```

CON'T FROM PAGE 16-PTDOS TO CP/M FILE TRANSFER

```
        MOV C,A         ;SAVE NUMBER OF CHARS. IN C-REG
        LXI H,82H       ;POINT TO FIRST CHARACTER
GETNA:  MOV A,M
        INX H
        CPI ' '         ;LOOK FOR SPACE SEPARATING FILE NAMES
        JZ GOTNA        ;IF FOUND DO IT
        DCR C           ;KEEP UP WITH CHARACTER COUNT
        JZ ERR1         ;MUST HAVE TWO FILES GIVEN
        JMP GETNA
GOTNA:  MOV A,C         ;C-REG CONTAINS REMAINING COUNT
        CPI 9           ;SEE HOW LONG PTDOS FILE NAME IS
        JNC ERR2        ;IF LONGER THAN 8 CHARACTERS
        LXI D,PTNAM     ;POINT TO STORAGE FOR PTDOS FILE NAME
GOTN1:  MOV A,M
        STAX D          ;MOVE THE NAME
        INX H
        INX D
        DCR C           ;TILL ALL THE CHARACTERS ARE USED UP
        JNZ GOTN1
        XCHG            ;PUT ADDRESS IN H,L
        MVI M,'/'       ;APPEND A '/1 (CR)' TO
        INX H
        MVI M,'1'       ;PTDOS FILE NAME
        INX H
        MVI M,CR        ;WILL BE LOCATED ON DRIVE 1
;
; ENABLE ERROR LEVELS 1,2 RETURNS FROM PTDOS
; DON'T ENABLE LEVEL 0 ERRORS-BAD STUFF
;
        LHLD SYSGLO     ;GET ADDRESS OF SYSTEM GLOBAL AREA
        LXI D,9         ;POINT TO START OF ERROR TRAP
        DAD D           ;ADDRESSES
        MVI A,0FFH      ;ENABLE LEVEL 2 TRAP RETURN
        MOV M,A
        INX H
        MOV M,A
        INX H
        MOV M,A         ;ENABLE LEVEL 1 TRAP RETURN
        INX H
        MOV M,A
;
; SET UNIT 1=DEFAULT UNIT FOR PTDOS
;
        MVI A,1         ;SPECIFY UNIT 1
        CALL SYS
        DB 12           ;SET UNIT AS DEFAULT
        JMP PTERR       ;IF AN ERROR IS DETECTED
;
; OPEN PTDOS FILE
; NOTE: THE FOLLOWING DOES NOT CHECK FOR FILE TYPE
;
        LXI H,0         ;STATIC BUFFERING
        LXI D,PTNAM     ;POINT TO FILE NAME
        CALL SYS
        DB 1            ;OPEN THE FILE
        JMP PTER1       ;ERROR IF NOT PRESENT
        STA FNUM        ;SAVE THE FILE NUMBER
        LXI H,0         ;ZERO NUMBER OF CHARACTERS
        SHLD PTCNT      ;RECEIVED FROM PTDOS FILE
;
; CPM FILE NAME PARSED INTO FCB AT 005CH BY CCP
; NO NEED TO FOOL WITH IT OTHER THAN TO REQUIRE IT LIVE
; ON DRIVE A AND TO INITIALIZE A FEW OF THE PARAMETERS
;
        MVI A,1
        STA FCB         ;SET CPM DRIVE=DRIVE A
        XRA A           ;SET REST OF CPM PARAMETERS
        STA FCB+12      ;TO ZERO
        STA FCB+14      ;SET S2=0
        STA FCB+15      ;SET CURRENT EXTENT=0
        STA EOF         ;CLEAR END-OF-FILE FLAG
```

```
;
; TRY TO OPEN CPM FILE. IF IT ALREADY EXISTS THEN
; ASK USER IF OK TO DELETE IT. IF OK TO DELETE THEN
; DELETE OLD FILE AND CREATE A NEW FILE
; ELSE CREATE THE FILE AND THEN OPEN IT
;
        LXI D,FCB       ;POINTER TO FILE NAME IN D,E
        MVI C,OPEN      ;OPEN COMMAND IN C-REG.
        CALL BDOS
        ORA A           ;IF COMES BACK 0FFH THEN NOT THERE
        JM CRETE        ;IF NEGATIVE THEN CREATE AND OPEN
        LXI D,MSG2      ;PRINT MESSAGE STATING THAT CPM
        MVI C,PSTRIN    ;FILE ALREADY EXISTS
        CALL BDOS       ;ASK IF WANT TO DELETE
        CALL ASKOK      ;GET RESPONSE
        JNZ QUIT
        LXI D,FCB       ;POINT TO FILE CONTROL BLOCK
        MVI C,13H       ;TELL CPM TO DELETE IT
        CALL BDOS       ;AND DO IT
CRETE:  MVI C,CREATE    ;FILE IS NOT THERE SO CREATE IT
        LXI D,FCB       ;POINT TO FILE NAME
        CALL BDOS
        ORA A
        JM ERR3         ;IF MINUS THEN CAN'T CREATE
        LXI D,FCB       ;- POSSIBLE DISK OR DIRECTORY FULL
        MVI ~~C,OPEN~~      ;~~TELL CPM TO OPEN~~ CREATED FILE DONE
        ORA A           ;IF A-REG CONTAINS A 0FFH ON RETURN
        JM ERR3         ;HAVE PROBLEMS OPENING FILE
;
; NOW BEGINS THE ACTUAL FILE TRANSFER
; PTDOS TEXT WILL BE READ INTO A BUFFER AT PTBUF
; CPM TEXT WILL BE STORED AT THE DEFAULT DMA ADDRESS 80H
;
; FIRST READ IN A BLOCK OF TEXT FROM PTDOS
;
        CALL RDPT       ;READ IN A BLOCK
        LXI H,PTBUF     ;SET PT POINTER TO START OF BLOCK
        SHLD PTADD      ;AND SAVE IT
        LXI H,80H       ;POINT CPM'S POINTER TO 80H
        SHLD CPMAD      ;AND SAVE IT
PROCES:
        CALL GETCH      ;GET A CHARACTER
        CPI ' '         ;IS IT A SPACE
        JZ SPACE        ;IF SO THEN PROCESS IT
PROCS1:
        CPI CR          ;IS IT A CARRIAGE RETURN
        JZ EOL          ;IF SO THEN NEED TO ADD A LINE-FEED
        CPI ';'         ;IS IT A SEMICOLON
        JZ PASSOV       ;IF SO THEN PASS-OVER REST OF LINE
        CPI 27H         ;IS IT A '
        JZ PASSOV       ;IF SO THEN PASS OVER REST OF LINE
        CPI 22H         ;ALSO FOR A "
        JZ PASSOV
        CPI LF          ;IGNORE ALL LINE-FEEDS, WE WILL
        JZ PROCES       ;INSERT THEM
        CALL PUTCH      ;OTHERWISE PUT THE CHARACTER IN CPM'S
        JMP PROCES      ;BUFFER AND GET NEXT CHARACTER
;
; PROCESSING FOR SPACES
; IF TWO OR MORE CONSECUTIVE SPACES THEN REMOVE AND
; REPLACE WITH A TAB
;
SPACE:  CALL GETCH      ;SEE IF NEXT CHARACTER IS A SPACE
        CPI ' '
        JZ SPCOU        ;IF SO THEN COMPRESS INTO A TAB
        PUSH PSW        ;OTHERWISE SAVE CHARACTER
        MVI A,' '       ;REPLACE ORIGINAL SPACE
        CALL PUTCH      ;IN CPM'S OUTPUT
        POP PSW         ;AND THEN ADD IN PRESENT CHARACTER
        JMP PROCS1      ;BACK TO NORMAL PROCESSING
;
; INSERT A TAB AND IGNORE FOLLOWING SPACES
```

CON'T FROM PAGE 17-PTDOS TO CP/M FILE TRANSFER

```
;
SPCOU:  MVI A,Ø9H       ;PUT IN THE TAB
        CALL PUTCH      ;PUT IN CPM'S OUTPUT
SPCO1:  CALL GETCH      ;GET NEXT CHARACTER
        CPI ' '         ;KEEP LOOPING TILL NO MORE SPACES
        JZ SPCO1
        JMP PROCS1      ;BACK TO NORMAL PROCESSING
;
; PROCESSING FOR END OF LINE (CARRIAGE-RETURN)
;   NEED TO SEND CR AND ADD A LINE-FEED
;
EOL:    CALL PUTCH      ;PUT IN CARRIAGE RETURN
        MVI A,LF        ;AND THEN INSERT A LINE-FEED
        CALL PUTCH
        JMP PROCES      ;BACK TO NORMAL PROCESSING
;
; THE FOLLOWING TRANSMITS THE REMAINDER OF THE LINE
; WITH NO MODIFICATIONS, THUS PREVENTING UNWANTED
; COMPRESSION OF SPACES SUCH AS WITHIN A DB '  '
; NOTE THAT SPACES FOLLOWING A  MVI A,'X'  TYPE CONSTRUCT
; WILL NOT BE COMPRESSED.
;
PASSOV:
        CALL PUTCH      ;PUT THE CHARACTER TO CPM
        CALL GETCH      ;GET THE NEXT FROM PTDOS
        CPI CR          ;LOOK FOR THE END OF THE LINE
        JZ EOL          ;IF FOUND, ADD A LINE-FEED
        JMP PASSOV      ;OTHERWISE, KEEP ON TRUCKIN
;
; THE FOLLOWING ROUTINE GETS A CHARACTER FROM PTDOS
; FIRST THE PRESENT CHARACTER COUNT IS TESTED TO
; DETERMINE IF WE STILL HAVE SOMETHING IN THE BUFFER.
; IF NOTHING IN THE BUFFER THEN WE WANT TO READ IN A
; BLOCK OF DATA.
;
GETCH:  LHLD PTCNT      ;GET PT'S CHARACTER COUNT
        MOV A,L
        ORA H
        CZ RDPT         ;IF ZERO THEN READ IN SOME MORE
        DCX H           ;REDUCE THE COUNT BY ONE
        SHLD PTCNT      ;AND SAVE IT.
        LHLD PTADD      ;GET THE ADDRESS OF THE CHARACTER
        MOV A,M         ;AND GET THE CHARACTER INTO THE A-REG
        INX H           ;BUMP ADDRESS FOR NEXT GETCH
        SHLD PTADD      ;SAVE IT
        RET             ;AND RETURN WITH CHAR. IN A-REG
;
; THE FOLLOWING ROUTINE PLACES THE OUTGOING CHARACTER
; INTO CPM'S DATA BUFFER. FIRST IT CHECKS IF THE DATA BUFFER
; IS FULL (128) CHARACTERS. (ADDRESS 8ØH + 128)=ADDRESS 1ØØH
; IF THE BUFFER IS FULL IT IS WRITTEN TO THE CPM FILE ON
; DRIVE Ø.
;
PUTCH:  MOV B,A         ;SAVE CHARACTER IN THE B-REG
        LHLD CPMAD      ;GET ADDRESS OF CPM DATA
        MOV A,H         ;SEE IF REACHED ADDRESS 1ØØH YET
        CPI 1
        CZ WRCPM        ;IF WE HAVE THEN WRITE BLOCK OUT
        MOV M,B         ;PUT CHARACTER IN BUFFER
        INX H           ;BUMP ADDRESS
        SHLD CPMAD      ;AND SAVE IT
        RET             ;KEEP ON TRUCKIN
;
; THE FOLLOWING ROUTINE WRITES A BLOCK OF DATA TO CPM
; (ASSUMED ON DRIVE Ø).
; AFTER THE DATA IS WRITTEN, THE DATA ADDRESS IS RESET
;
WRCPM:  PUSH B          ;SAVE CHARACTER
        MVI C,WRITE     ;ISSUE WRITE COMMAND
        LXI D,FCB       ;POINT TO FILE CONTROL BLOCK
        CALL BDOS       ;DO IT
```

```
        POP B           ;GET CHARACTER BACK
        LXI H,8ØH       ;RESET DATA BLOCK ADDRESS
        RET
;
; READ A BLOCK OF DATA FROM PTDOS (DRIVE 1)
;
RDPT:   LDA EOF         ;SEE IF END-OF-FILE WAS FOUND
        ORA A           ;LAST TIME
        JNZ ALLDON      ;IF SO, THEN WE ARE ALL DONE
        LDA FNUM        ;ELSE GET FILE NUMBER
        LXI B,2ØØH      ;READ IN 2ØØH BYTES  (WHY NOT?)
        LXI D,PTBUF     ;POINT TO WHERE IT GOES
        CALL SYS        ;TELL PTDOS ABOUT IT
        DB 3
        JMP MABEOF      ;IF HERE, CHECK IF END-OF-FILE
RDPT1:  CALL NEGBC      ;IF HERE, NOT END OF FILE
        LXI H,2ØØH      ;COMPUTE HOW MANY CHARACTER READ IN
        DAD B
        SHLD PTCNT      ;SAVE THE BYTE COUNT
        MOV A,H         ;IF GOT ZERO BYTES THEN DONE
        ORA L
        JZ ALLDON
        LXI H,PTBUF     ;RESET PT DATA BUFFER ADDRESS
        SHLD PTADD
        LHLD PTCNT      ;RETURN WITH COUNT IN H,L
        RET
;
; TEST IF END-OF-FILE REACHED
;
MABEOF:
        CPI 18H         ;IS IT END OF FILE
        JNZ PTERR       ;NOT AN END-OF-FILE
        STA EOF         ;SET END-OF-FILE FLAG
        JMP RDPT1       ;CONTINUE AS THOUGH NO END-OF-FILE
;
; NEGATE B,C REGISTER PAIR
;
NEGBC:  MOV A,C
        CMA             ;COMPLIMENT C-REG
        MOV C,A
        MOV A,B
        CMA             ;COMPLIMENT B-REG
        MOV B,A
        INX B           ;INCREMENT B,C
        RET
;
; DATA TRANSFER HAS BEEN COMPLETED FROM PTDOS
; NOW NEED TO FILL THE REMAINDER OF CPM'S TEXT BUFFER
; WITH 1A'S (CONTROL-Z). (CPM USES 1A'S TO SIGNIFY END OF
; TEXT).
ALLDON:
        LHLD CPMAD      ;GET END OF CP/M'S DATA BUFFER
ALL1:   MOV A,H         ;TEST FOR 1ØØH ADDRESS
        CPI 1
        JZ ALLD1        ;WHEN THERE, ALLLL DONE
        MVI M,1AH       ;FILL IN A 1A
        INX H           ;NEXT LOCATION
        JMP ALL1
;
; WRITE LAST CPM DATA BLOCK TO DISK, CLOSE FILES
; AND GO HOME
;
ALLD1:  MVI C,WRITE     ;WRITE BLOCK TO DISK
        LXI D,FCB
        CALL BDOS
        MVI C,CLOSE     ;CLOSE CPM FILE
        LXI D,FCB
        CALL BDOS
        LDA FNUM        ;CLOSE PTDOS FILE
        CALL SYS
        DB 7
        JMP PTERR
```

CON'T FROM PAGE 18-PTDOS TO CP/M FILE TRANSFER

```
QUIT:   MVI A,0C3H      ;PUT IN THE JUMP AT 0 WHICH
        STA 0           ;PTDOS MESSES UP
        JMP 0           ;AND GO BACK TO CPM
;
; PRINT ERROR MESSAGE FROM PTDOS WITH ERROR NUMBER
;
PTERR:  PUSH PSW        ;ERR NUM IN ACC
        LXI D,ERRMS1    ;PRINT MESSAGE PTDOS ERROR #
        MVI C,PSTRIN
        CALL BDOS
        POP PSW
        CALL NMOUT      ;PRINT THE ERROR NUMBER
        CALL CRLF       ;DO A CR-LF
        JMP ALLDON      ;QUIT
;
PTER1:  LXI D,MSG3      ;PRINT MESSAGE STATING PTDOS
        MVI C,PSTRIN    ;FILE NOT FOUND
        CALL BDOS
        JMP QUIT        ;QUIT IF PTDOS FILE DOSEN'T EXIST
;
; PRINT A CARRIAGE RETURN, LINE-FEED ON CONSOLE
;
CRLF:   LXI D,CRLFM
        MVI C,PSTRIN
        JMP BDOS
;
; ROUTINE TO ASK USER FOR AN AFFIRMATIVE RESPONSE
;
ASKOK:  MVI C,CONIN     ;CONSOLE INPUT
        CALL BDOS
        PUSH PSW        ;HOLD CHARACTER WHILE WE
        CALL CRLF       ;GENERATE A CR-LF
        POP PSW         ;GET OUR CHARACTER BACK
        CPI 'Y'         ;Y IS AN AFFIRMATIVE ANSWER
        RZ
        CPI 'y'         ;y IS ALSO AFFIRMATIVE
        RZ
        CPI CR          ;AS WELL AS A CARRIAGE RETURN
        RET
;
; PRINT HEX CONTENTS OF THE A-REG
;
NMOUT:  PUSH PSW
        RAL
        RAL
        RAL
        RAL
        CALL NIBOU
        POP PSW
NIBOU:  ANI 0FH
        CPI 10
        JC NIB1
        ADI 7
NIB1:   ADI 30H
        MVI C,CONOUT
        MOV E,A
        CALL BDOS
        RET
;
; ACTUAL MESSAGES
;
CRLFM:  DB LF,CR,'$'
ERRMS1: DB 'PT ERROR #  $'
ERMS1:  DB 'EXPECTING FILE NAME',CR,LF,'$'
ERMS2:  DB 'INVALID PTDOS FILE NAME',CR,LF,'$'
ERMS3:  DB 'DISK FULL!!',CR,LF,'$'
MSG3:   DB 'PTDOS FILE DOES NOT EXIST',CR,LF,'$'
MSG2:   DB 'OUTPUT FILE EXISTS. DELETE? $'
MSG1:   DB '+-+- DOWNLOAD PROGRAM -+-+',CR,LF
        DB 'PLACE CPM DISK IN DRIVE A(0)',CR,LF
        DB 'AND PTDOS DISK IN DRIVE B(1)',CR,LF
```

```
        DB 'HIT RETURN TO CONTINUE $'
ERR1:   LXI D,ERMS1     ;NO FILE NAME GIVEN
        MVI C,PSTRIN
        CALL BDOS
        JMP QUIT
ERR2:   LXI D,ERMS2     ;FILE NAME TOO LONG
        MVI C,PSTRIN
        CALL BDOS
        JMP QUIT
ERR3:   LXI D,ERMS3     ;DISK FULL, CAN'T CREATE NEW FILE
        MVI C,PSTRIN
        CALL BDOS
        LDA FNUM
        CALL SYS
        DB 7
        JMP PTERR
        JMP QUIT
;------------------
; PTDOS EQUATES
;------------------
SYS:    EQU 0BCBCH      ;PTDOS SYSTEM ENTRY POINT
SYSGLO: EQU 0BCA5H      ;ADDRESS OF POINTER TO SYS GLOBAL
CR:     EQU 0DH         ;CARRIAGE RETURN
LF:     EQU 0AH         ;LINE-FEED
;--------------------------
; CPM EQUATES
;-------------------------
BDOS:   EQU 5   ;ENTRY POINT FOR BDOS
FCB:    EQU 5CH ;ADDRESS OF CPM DEFAULT FILE CONTROL BLOCK
CONIN:  EQU 1   ;CONSOLE INPUT
CONOUT: EQU 2   ;CONSOLE OUTPUT
PSTRIN: EQU 9   ;PRINT $ TERMINATED STRING
OPEN:   EQU 15  ;OPEN FILE
CLOSE:  EQU 16  ;CLOSE FILE
WRITE:  EQU 21  ;WRITE FILE SEQUENTIAL
CREATE: EQU 22  ;CREATE FILE
;-----------
; DATA AREA
;-----------
EOF:    DB 0    ;FLAG FOR END OF FILE
FNUM:   DB 0    ;PTDOS FILE NAME
PTADD:  DW 0    ;STORAGE FOR ADDRESS OF PTDOS READ BUFFER
CPMAD:  DW 0    ;STORAGE FOR ADDRESS OF CPM WRITE BUFFER
PTCNT:  DW 0    ;CHARACTERS REMAINING IN PTDOS BUFFER
PTNAM:  DS 10H  ;LOCATION OF PTDOS FILE NAME
        DS 60H  ;SAVE A LOT OF ROOM FOR STACK
STACK:  EQU $   ;SET THE STACK HERE
        DS 10   ;ALLOW A LITTLE ROOM
PTBUF:  EQU $   ;LOCATION OF PTDOS INPUT BUFFER
        END
```

GAMEPAC-1 DOESN'T RE-ACT:

"...I am able to run everything (Gamepac-1, Gamepac-2 & TREK-80) on the tape except those items that make up Gamepac-1. These programs load and execute except I cannot get any re-action to input from my terminal.

I have an IMSAI with CUTS, an ADM-3 which I use for input and display. In addition I have a monitor and a VDM-1 PTC I/O and I have historically had problems since many parograms do not use th I/O routines of the operating system.

The displays appear as expected on the monitor. Input echos on the monitor, but has no effect. Do you have any ideas or suggestions?

Thank you for your assistance,

John E. Breden
921 Waterview Circle
Richardson, Texas 75080

SLAC PASCAL PATCHES & BENCHMARKS

by Mike McKelvey

I am writing with the solution to a problem I raised in the April/May 1980 issue of Proteus News and also to present some benchmark results based on the program in the August 1981 Interface Age.

Back in that April/May issue of Proteus News, I wrote about the problem of PTDOS not normally printing the underline character on the screen. Helios library disk H-1 contains a little program INITPATB which alters some memory values in PTDOS and fixes the problem. The trouble is that INITPATB only works with PTDOS 1.4. I sent in the source code and asked if anyone knew how to modify it for PTDOS 1.5.

I just discovered how yesterday. I tried using the Debugger to examine memory in the areas of the changed values. You can use the Debugger without loading a program to debug. I set the Debugger so it would display 8080 instructions for the memory values of the area of PTDOS 1.4 that was altered. I saved this information in a disk file and then printed it out on the printer. I did the same thing for PTDOS 1.5. Then I looked for patterns of instructions in PTDOS 1.5 that were similar to those that were altered in PTDOS 1.4. This way I found the equivalent memory locations for PTDOS 1.5 and I could substitute them in the program. The revised program follows the letter.

If you are familiar with the use of EDIT but not ASSM, type the program in using EDIT and call it INITPATS. Then, after you get the PTDOS prompt again type:

```
*ASSM INITPATS,,INITPATB
```

Now, use EDIT again and add INITPATB to the START.UP file. This will cause the the corrections to be made automatically each time you boot-up PTDOS.

This may seem like a trivial thing, but it sure felt good to be able to solve the problem now when I had no idea how to do it when I wrote before.

On to benchmarking. Recently, I've been interested in comparing the speed of the same program in different languages. This started when I ran a BASIC solar energy program from a recent BYTE. What seemed like a simple program took over 4 minutes to run. I had been waiting for a reason to try out Processor Technology FORTRAN and this seemed like a good opportunity. I translated the program into FORTRAN, sure that it would run in about 1/4 the time since the FORTRAN is a compiler.

The FORTRAN was only 10 seconds faster than BASIC! I still have trouble believing it. One of my reasons for getting FORTRAN was that I thought it offered great speed compared to BASIC. Some things about it are nice compared to PTC BASIC like long variable names. They make a program much more understandable.

Now I wanted to make more speed comparisons. The August 1981 Interface Age contains an article giving the results of 70 tests of a BASIC benchmark program on various micros. A Sol with Extended Cassette Basic came in at 1812 seconds. Not too great, but better than some.

When I tried the program, as published, with Extended Disk Basic it wouldn't run because of the abnormal exiting from the for-next loop. It did run after modification.

Then I translated to program into FORTRAN, FOCAL and Slac Pascal. FORTRAN did a little better this time. FOCAL took almost 3 times as long as Basic and, with my translation, produced some errors. Slac Pascal was the standout at about 1/5 the time of FORTRAN.

The results of my testing are as follows:

BASIC	1863 seconds
FORTRAN	1227 seconds
FOCAL	5166 seconds
Slac Pascal	248 seconds

This made me wish, more than ever, that I could use Slac Pascal with real numbers and that I could get the 64k version working. When I try reassembling the interpreter and the RUN program for 64k, they don't work. I would be nice if someone who has done it could tell me what in the source code, on H-2, has to be changed.

I'll send a copy of INITPATB to the Helios Library along with a revised Sol2 printer driver that allows the use of 88 character print wheels, of which there is a much wider type selection. It Won't try to print one of the missing characters.

Cordially,

Mike

Michael A. McKelvey
330 S. State Street
Ann Arbor, MI 48104

INITPATS UNDERLINE PATCH

```
 TITL I/O INITIALIZER FOR NEW SLAC PTDOS DOS1.4.1 6/05/78
* MODIFIED 9/4/81 TO WORK WITH PTDOS 1.5 BY M. McKELVEY
 COPY NPTDEFS
 ORG 0100H
 XEQ BEGIN
BEGIN MVI A,0  NOP
 STA 0B715H  ZAP CONIN RZ WHICH SNAGS DELETES
 MVI A,7FH BACKSPACE CHAR
 STA 0B5F2H  CONIN BACKSPACE
 STA 0B63AH  CONIN ECHOES FOR BS
 STA 0B73EH  VDM USES FOR BS
 CALL SYS
 DB RETOP
 END
```

BASIC BENCHMARK PROGRAM

```
100 REM.  INTERFACE AGE's benchmark program to
110 REM.  'discover the first 1000 prime numbers
120 REM
130 PRINT "Starting:"
140 FOR N=1 TO 1000
150 FOR K=2 TO 500
160 LET M=N/K
170 LET L=INT(M)
180 IF L=0 THEN  LET K=500
190 IF L=1 THEN 220
200 IF M>L THEN 220
210 IF M=L THEN  LET K=500
220 NEXT K
230 IF M<>L THEN  PRINT N;
240 NEXT N
250 PRINT CHR(7)
260 PRINT "Finished."
270 END
```

CON'T FROM PAGE 20-SLAC PASCAL PATCHES & BENCHMARKS

FORTRAN BENCHMARK PROGRAM

```
C          INTERFACE AGE's benchmark program to
C          'discover the first 1000 prime numbers
C
      REAL K,L,M,N
      TYPE 'Starting:'
      DO 240 N=1,1000
      DO 220 K=2,500
      M=N/K
      L=INT(M)
      IF (L.EQ.0) GO TO 230
      IF (L.EQ.1) GO TO 220
      IF (M.GT.L) GO TO 220
      IF (M.EQ.L) GO TO 240
  220 CONTINUE
  230 WRITE (1,235) N
  235 FORMAT (I4,Z)
  240 CONTINUE
      TYPE '\07\'
      TYPE 'Finished.'
      STOP
      END
```

PASCAL BENCHMARK PROGRAM

```
PROGRAM BENCHMARK (OUTPUT);

(*************************************************************
*                                                           *
*           INTERFACE AGE's benchmark program to            *
*           'discover the first 1000 prime numbers          *
*                                                           *
*************************************************************)

LABEL 220, 230, 240;
VAR  K,L,M,N: INTEGER;

BEGIN
WRITELN ( 'Starting:' );
FOR N := 1 TO 1000 DO
      BEGIN
            FOR K := 2 TO 500 DO
            BEGIN
                   M := N DIV K;
                   IF N < K THEN GOTO 230;
                   IF M = 1 THEN GOTO 220;
                   IF N MOD K = 0 THEN GOTO 240;
       220: END;
       230: WRITE (N:4);
 240: END;
WRITELN ( CHR (103) );
WRITE ( 'Finished.' );
END.
```

FOCAL BENCHMARK PROGRAM

```
2.1  C  INTERFACE AGE's benchmark program to
2.2  C  'discover the first 1000 prime numbers
2.3  C

3.01 SET N = 0
3.02 SET N = N+1
3.03 IF (1000-N) 6.1
3.04 SET K = 1
3.05 SET K = K+1
3.06 IF (500-K) 3.13
3.07 SET M = N/K
3.08 SET L =FITR(M)
3.09 IF (L) 9.9,3.13,3.10
3.10 IF (L-1) 3.13,3.12,3.11
3.11 IF (M-L) 9.9,3.14,3.12
3.12 GOTO 3.05
3.13 TYPE %4.00,N
3.14 GOTO 3.02

6.1 TYPE "BELL",!
6.2 TYPE "Finished."
6.3 QUIT
```

REGARDING RICHARD BJORNDAL'S RANDOMIZE FOR PT BASIC
BY MURRAY MACKENZIE

I was unaware of any need to implement a randomize function for PT. BASICS. I am unable to find any repetition in the RND function providing only the RND(0) is used. Once "RND" is given a numerical argument other than "0", anywhere in the program, a seed chain is established and even RND(0) becomes predictable thereafter. Richard's mini-program will not fix it. Re-running the program will not fix it. Scratching and re-loading the program will not fix it.

Removing any statement in the program that gives a numerical argument other than "0" to RND will fix it.

Example: Try this.....

```
10 for I=1 to 6:print RND(0);:next I
   (Run that programI several times, observing randomness)
        .....now add this line..
20 Print RND(4)
   (Run againI, several times..)
```

On the second run, after RND(4) has been "seen", the RND(0) function becomes repetitive. Remove 20 and run again and all is random. .... Or am I, missing a point?

However, Applesoft and the BASICS on "Softcard" can make good use of Richard's program. They require seeding.

Murray Mackenzie
38 Inniswood Drive,
Scarborough, Ontario
CANADA M1R 1E5

Editor's Note:

The problem with use of RND(O) is that the next time you load BASIC and run your program it will behave exactly the same as it did the last session, since the number generator will start again at the beginning. Seeding allows you to get a different series of random numbers in each unrelated session. But not all numbers are good as seeds. Depending upon the pseudo-randam number generator algorithm, certain seeds will produce a relatively short series of unique numbers before beginning to repeat. Perhaps someone who has the BASIC source code (Proteus P12) and who knows more about random number generators could make some recommendations for getting good seeds. ..

MY DELAY LINE FAILED
Robert A. Ellingsworth

I have been a member of PROTEUS for almost two years and find the newsletter extremely valuable. I wish to pass along a couple of possible 16KRA delay line solutions and, of course, some questions of my own.

The delay line is apparently a non-standard part made by Data Delay Devices (385 Lakeview Ave., Clifton, NJ 07011). The item is a 74S04 hex inverter that has various inductance-capacitance sections which determine the delay. The required delays are 100, 150, 250, and 350 nanoseconds (see para 5.3.3 of the Users Manual). The Data Delay Devices product line includes a number of delay lines with ten sections. The DDU-5J-10500 has 10 sections with 50ns. delay each. By using the second, third, fifth, and seventh taps, the desired delay could be realized. Similar results could be obtained with P/N DTTLDM-500 from Engineered Components Company, 358 Sacramento Drive, San Luis Obispo, CA 93401. Neither will physically fit in the PCB socket, but a module could be made using a DIP header. (Note that the top buss slot allows extra height.)

I was not familiar with the above mentioned product lines when my delay line failed. Instead mine was fixed using a "brute force" method which utilizes four 74L04s and a DIP header. The inverters are connected in series to obtain the required delays and mounted on the header. The delay must be checked with a dual trace scope or interval timer. An even number of inverters must be used for each section so that the signal remains in phase. The packages can be piggybacked with all pins 7 and 14 soldered together and to the header. By using the bottom IC as the last in each leg, connections to the header can be simplified since pins 4, 6, 10, and 12 are gate outputs and correspond to pins 4, 6, 12, and 14 of the U71 socket if the IC is positioned with pin 1 corresponding to pin 1 of the socket. My 16KRA has been working for over a year with this fix installed. Use of some 74C04 inverters would help reduce the package count since they have a significantly longer propogation delay. I hope someone finds this information useful.

[Editor's note: Regarding the ECBASIC source, see my remarks in the "What's New" section of this issue. --Stan.]

I have received the ECBASIC source code that I ordered, but when I tried to find the tape routines (to compare them to the non-functional routines in the G-2 BASIC) I was disappointed that they are not included. In fact, I am having a problem finding a correspondence between the source and the object code. I would appreciate your comments.

I am planning to purchase a disk system next spring and am considering DISCUS, Vista, North Star, and Micropolis. I would appreciate hearing from any members who have used or tried to use one of these systems with a SOL having an SD Systems Expandoram I (4116 version). I have not had any of the problems with this board noted by other members, but maybe a disk system will bring out the worst.

I have an opportunity to purchase the PT Extended Disk FORTRAN. Is it possible to adapt it to a CP/M system or to the North Star or Micropolis disk operating systems? The portable PTDOS discussed in the APR/MAY issue of PROTEUS/News sounds like an alternative solution.

Along the same line, availability of the PASCAL (hopefully with floating point capability) from library disk H1 in a CP/M compatible form would be desirable.

If I understand the letter from Bruce Barron on page 17 of Vol.4, #1 correctly, a relocated SOLOS that is modified to emulate CUTER allows operation of ECBASIC, ALS-8 (with the possibly exception of the VDM driver), and most other non-game programs without modification. This makes total conversion to F000 much simpler.

I have seen a couple references to the TAD Enterprises modification to ECBASIC but a review in the newsletter would be appreciated.

Sincerely,
Robert A. Ellingsworth
Robert A. Ellingsworth

P.S. About a year ago I sent you some comments on my OKIDATA M Line 80 printer. I believe I stated that the printer did not directly interface to the SOL parallel port and required some extra circuitry. Since that time I have discovered that if the BUSY line is used instead of the ACK line that I had used to feed PDXR, no external circuitry is required.

I have gotten good service out of the printer but I reqret not getting the model 82 with forms control and bidirectional printing.

Robert A. Ellingsworth
1134 N. Stark St.
Davenport, Iowa 52804
October 19, 1981

WHAT'S NEW?

ECBASIC Source Code Really Isn't

Isn't what, you ask? It isn't the ECBASIC source code! Sometime after distributing the ECBASIC source code, we discovered that it was not the source for the released version of BASIC which we were told it was. In fact, it was an early in-progress version of BASIC that was being developed by PTC from the BASIC/5 code. It is missing major portions of the features of ECBASIC, such as cassette I/O.

When Tom Digate, one of our members, pointed this out to me, he and I arranged for him to try altering the enhanced version of Extended Disk BASIC (known as Business BASIC Level 1) to work on CP/M disk. The idea was that most members buying the ECBASIC source probably had or would eventually get a disk system. Tom, you may recall, was the author of a modification that let ECBASIC object tapes be converted to run as a disk BASIC under CP/M. So, he was the natural one to ask for help on the project.

Tom has recently contacted me, saying that he has done the majority of the conversion and is almost ready to distribute it. He and I are working out the legalities of licensing the distribution, under Proteus's license from Processor Technology. Even though the company is out of business, it technically still owns the source code and someday someone may legally pick up that asset. We have to be careful.

I'll announce our arrangement for distribution in the next issue of Proteus News. Purchasers of the deficient source code will be given full credit toward the purchase of the highly improved code.

VULCAN Database Manager Gives Birth to dBASE II

Way back there in the past issues of Proteus, I mentioned a database manager that ran under PTDOS on the Helios. At the time, it was not very complete, and I wasn't impressed.

Recently, I bought the manual for dBASE II, the widely advertised "Relational" database management system for CP/M systems. It sure bore a striking resemblence to VULCAN, I thought, but it is much improved. And what do ya know, it is exactly that. Now, instead of sequential files, dBASE uses either sequential or indexed files, with the indices organized in the B-tree method. Lots of other features have been added, too.

In my opinion, dBASE II is only a pseudo-relational database manager, since the data not only appears to be in tabular form, it is also stored that way, redundantly. That is, if two database files are defined with variables in common, the data is stored in both files. In my mind, a true relational database manager will store the data only in one place and refer to it by pointers. That way, changing it in one file changes the other, too.

Even with the short comings, dBASE II is still quite a handy system, and much more powerful than ordinary mailing-list programs. In fact, it is actually a programming language with high-level file management and report generation commands.

WHERE TO FIND DISKS & RIBBONS
Mike McKelvey

September 14, 1981

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

Dear Stan:

I recently received the latest issue of Proteus/News and was excited about the idea of making Sol disappear. That sure sounds more flexible than the other schemes of moving it to F000. It was also interesting to read about your smoked Helios. I too have been ignoring that mandatory update notice.

My main reason for writing is to let you and the other members know about a source of inexpensive floppy disks. I don't mean to harm Lewis Moseley and his attempts to sell no-name disks for $3.00, but I think this is a better deal.

Communications Electronics, here in Ann Arbor, is selling Memorex disks for the Helios for $2.19 each in quanities of 100. Smaller quanities are 10% more or about $2.40. I know the owner of the company, Ken Asher. He's a good guy. He keeps about 200,000 disks in stock and the prices are low because of that volume. He told me they may even be coming down if he can get a better price from Memorex.

The Memorex disks for the Helios are Part no. 3201-3015. Many other kinds are available and the price may vary from those for the Helios. Communications Electronics' toll free phone number is (800) 521-4414.

I was using 3M disks before that I got for $3.00 from AB Computers in Pennsylvania. The differences with the Memorex that I have noticed are that they have a more rigid jacket than the recent 3M disks (maybe good). The Memorex disks also make more noise when spinning around.

As long as I'm on the subject of cheap supplies, you might try Timberline Information Processing Supplies for Diablo ribbons at 4.16 each in a box of 12. Buy 2 dozen and get a free plastic daisy wheel. I also get paper from them. A box of 2800 sheets of heavy weight 9 1/2 x 11 plain white (tears to 8 1/2) is $25.96. I seen it for over $40.00 from other sources. They have good prices on mailing labels too.

Timberline's toll free phone number is (800) 547-5743. Call between 7:00 am and 5:00 pm Pacific time. They ship from Portland or Indianapolis, whichever is closer to you.

Thanks for what you do for me and the other members of Proteus.

Cordially,

Mike McKelvey

FIX FOR VISTA CBIOS

Although my Vista V-200 disk system has been a reliable performer, it has had an annoying tendency to repeat characters entered from the keyboard when operating under CP/M. Since the effect went away as soon as I loaded Basic, I reasoned that the problem might be with the CBIOS that Vista furnishes for the SOL and called the SOLBIOS by them. I compared the I/O routine with one by Fr. McGahee and made the following changes to the CONSOLE INPUT routine:

```
AS GIVEN        CINP    IN      STATP
                        ANI     IRDY
                        JNZ     CINP
                        IN      DPORT
                        ANI     07FH
                        RET
                        DS      16

CHANGE TO       CINP    IN      STATP
                        CMA
                        ANI     IRDY
                        JZ      CINP
                        IN      DPORT
                        ANI     07FH
                        RET
                        DS      16
```

Since the two routines seem to be the same logically, perhaps some 8080 maven can explain why the change works. In any case the fix has ended any tendency toward keyboard bounce.

Leonard Kalish
580 S. San Vicente Blvd. #3
Los Angeles, CA 90048
(213) 653-6874

Is CP/M 2.2 for Helios Out There?

Has anyone adapted CP/M 2.2 for the Helios yet? Lifeboat Associates, as far as I know, has adapted CP/M 1.4 but has no plans for 2.2 Has anyone else done it?

24

ATOR: A PASCAL CONVERSION ROUTINE
By Daniel S. Hunt

There are many times that a real number cannot be input directly into a real variable, but first must arrive as a string of ascii characters. Typical situations: the number occurs in an edited input line or the number is an ascii token scanned from an input string, such as might be found in an interpreter. In either case, the ascii number must be converted to real machine-oriented representation prior to arithmetic operations.

BASIC accomplishes the conversion with the built-in VAL function, as in:

```
             5 DEFSNG X
            10 INPUT "NUMBER"; NUMSTR$
            20 X = VAL(NUMSTR$)
```

Pascal has no such routine in its definition. Without one, you must cast your i/o algorithms around the standard Pascal procedures, READ(X) or READLN(X). This can be quite restrictive if your implementation of Pascal provides "unfriendly" editing routines for console or file i/o. Some Pascals, for instance, do not allow you to delete and replace a bad character as you execute a READLN(X). Make a mistake during response to a READLN and the computer may even hang!

The function ATOR (ascii to real) may therefore have a place in your Pascal library. You'll find yourself using it quite often, particularly with input parsing and user console routines.

The version I've written here emulates Microsoft BASIC's VAL function fairly closely. VAL ignores preceding blanks in the input string. It converts only up to the first non-numeric character in the string. The sequence "hbb-123.4AB55" (the small b's represent blanks) will convert to a real number of -123.4. The sequence "Not a number 1010.1" converts to 0.0.

In this example, ATOR uses as its input string a record built from a length integer and an array of characters. While there is no reason that the STRING type present in some versions of Pascal cannot be used, I use the do-it-yourself string to avoid the overhead of manipulating dedicated string types.

ATOR is machine and translator-independent in that it computes the resulting real value, rather than converting to real by use of knowledge about the manner in which real is represented in the computer. It makes no difference whether your Pascal's REAL type is floating point or BCD. If the compiler produces BCD reals, ATOR will show a slight lag due to the slowness of most BCD multiplication routines.

Before conversion begins, ATOR traps any zero length strings and returns 0.0; this section could be eliminated if it is known that no null length strings will be passed to ATOR from the program.

Leading blanks are caught in the following section at the same time the input string is scanned for minus or plus values. In the case of a find, the intrinsic procedure, MOVELEFT, is used to overlay the unsigned part of the string over the first position in the string, and the string length field is reduced by one character. If a minus sign is found, a boolean switch is set to true for later use.

Following a scan for imbedded blanks and alphas, in which furthur string length reduction may happen, the position of the decimal point, if any, is fixed by variable DPOS. DPOS is then used to count the number of characters to the right and left of the decimal point. Conversion then occurs in a repetitive set of multiplications against each character converted from ascii to its binary equivalent in PROCEDURE DECODEACHAR.

The supporting PROCEDURE READSTR provides console-edited input of the string to be fed to ATOR. It is the sort of line input routine that can be used by people unfamiliar with computers. Such a user need only be told to type characters, use the delete key if he goofs, and use the return key when everything looks all right. READSTR is customized to the SOL-20/VDM1 cursor routines, i.e., the delete key erases the character on the screen by moving the cursor left, writing a blank, and then moving the cursor left again. The simple editing method is preferable to input routines which follow the more complex control-character syntax of CP/M's user interface.

Daniel S. Hunt
822 Green Valley
Newbury Park, CA 91320

CON'T FROM PAGE 24-ATOR:A PASCAL CONVERSION ROUTINE

```
program atordemo;

const
    strlen = 80;
type
    stringtype = record
        len : integer;
        val : array[1..strlen] of char;
    end;
var
    s : stringtype;
    x : real;

procedure readstr(var s:stringtype);
var
    i : integer;
    ch : char;
begin

    {Blank string}

    s.len := 0;
    s.val[1] := ' ';
    move(s.val[1],s.val[2],strlen - 1);

    {Edit input until end of line}

    repeat
        read(ch);
        case ord(ch) of
        $0d: {End of line}
            EXIT;
        $7f : {Delete key}
            begin
                if s.len > 0 then
                    begin
                        {Blank deleted char in record}
                        s.val[s.len] := ' ';
                        s.len := s.len - 1;
                        {Cursor left, blank char on screen}
                        write(chr(1),' ',chr(1))
                    end
            end;


        else {A new character}
            if (ch >= ' ') and (s.len < strlen) then
            begin
                s.len := succ(s.len);
                s.val[s.len] := ch
            end
        end {case}

    until false; {Forever loop}
end; {readstr}


function ator(var s:stringtype) : real;
const
    minussign = '-';
    plussign = '+';
    asciizero = 48;
var
    ch : char;
    ptr, ncl, ncr, dpos : integer;
    realval,mult : real;
    minus : boolean;

    procedure decodeachar;
    begin
        realval := realval
                + ((ord(s.val[ptr]) - asciizero) * mult);
    end;{decodeachar}

begin {ator}

  {Trap null string}

  if s.len = 0 then
  begin
      ator := 0.0;
      EXIT
  end;

  realval := 0.0; minus := false;

  {Strip leading blanks & set minus switch}

  ch := s.val[1];
  while (ch = ' ') or ( ch = '-') or ( ch = '+') do
  begin
    if ch = '-' then
        minus := true;

    {Overlay position 2 string on position 1...}
    moveleft(s.val[2],s.val[1],s.len-1);

    {...and shorten length of string by one}
    s.len := s.len - 1;

    ch := s.val[1];
  end;

  ptr := 1; dpos := 0; {initialize}

  repeat
      ch := s.val[ptr];

      {Trap imbedded blanks and non-numerics}

      if not( ((ch >= '0') and (ch <= '9'))
          or (ch = '.') ) then
      begin
          s.len := pred(ptr);
          if s.len <= 0 then
          begin
              ator := 0.0;
              EXIT;
          end;
      end;

      {Position decimal point}

      if ch = '.' then dpos := ptr;

      ptr := succ(ptr);

  until ptr > s.len;


  {Count characters both to right and left of decimal point}

  if dpos = 0 then
  begin
      ncl := s.len;
      ncr := 0
  end
  else
  begin
      ncl := dpos-1;
      ncr := s.len - dpos
  end;

  { Convert characters left of decimal point }

  mult := 1;
  for ptr := ncl downto 1 do
  begin
      decodeachar;

      mult := mult * 10
  end;

  { Convert characters right of decimal point }

  mult := 0.1; ptr := dpos + 1;
  while ncr > 0 do
  begin
      decodeachar;
      mult := mult * 0.1;
      ncr := ncr - 1; ptr := ptr + 1;
  end;

  {Set sign of converted number}
  if minus then
      realval := -realval;
  ator := realval;

end;{ATOR}


begin {***MAIN***}

    repeat
        write('Enter string: ');
        readstr(s);
        x := ator(s);
        writeln(x)
    until false;

end.
```

```
A>atordemo
Enter string: 44.5
 .4450000E+02
Enter string:      44.5
 .4450000E+02
Enter string: a44.5
 .0000000E+00
Enter string: -44.5
-.4450000E+02
Enter string: -44.5ab cd end
-.4450000E+02
Enter string:
 .0000000E+00
Enter string:
```

Above: Demonstration run of "Atordemo", showing conversion of ASCII numbers with leading blanks, signs, and embedded letters into real numbers.

{ Daniel S. Hunt
822 Green Valley
Newbury Park,CA91320}

BUILDING THE 1978 REV B (NEW) SOL-PC

by Lynn A. McCroskey

SONICS
SONICS ASSOCIATES, INC.
237 OXMOOR CIRCLE
BIRMINGHAM, AL 35209
TELEPHONE: (205) 942-9631

October 5, 1981

Dear Dr. Sokolow:

There was a letter from Victor D. Bennight in the last Proteus Newsletter requesting assistance in assembling the 1978 revision "B" Sol P.C. I too was enticed into buying this unpopulated board by the price. Big old silly me. How did I know that the board was so different?

Fortunately, I have managed to resolve the differences and my 1978 Sol is running beautifully. These are the major areas where the P.C. is different:

1. C-75 - This is a bypass or pulse stretching capacitor. The value is unknown. My P.C. works fine without it and an examination of the circuit shows it to be in a non-critical area. If anyone should find out what this capacitor is, naturally, I would be interested in knowing.

2. C-78 - This is a .047mf power rail bypass capacitor.

3. R-91 - A 1.5k 5% 1/4 watt pullup resistor for the parallel input and load lines to U-76, the modified phantom IC. (More details to follow.)

4. R-160 - A 47 ohm 5% 1/4 watt - used in the serial input's opto-isolator circuit as described in one of the early "Access" newsletters.

5. U-114 - This is a resistor network which replaces the handful of pullup resistors used on earlier Sols. A Bourns 4116R-002-152 works fine and is available from Active Electronics, 133 Flanders Road, Westboro, Mass. 01581.

6. U-115 - Identical to U-144.

7. U-116 - Resistor network of a different value, Bourns 16-2-103. Available from Active Electronics.

8. U-76 - This was the biggest headache. It is not a 74LS175 as in earlier Sols. It was changed to a 74LS195 four bit shift register. The chip is used in the phantom circuit and is connected as shown on the modified schematic enclosed.

9. A jumper is required from pin 10 of U-45 to location "BB" next to U-47. Location "BB" is actually connected to pin 5. This connects "DBIN" to the output of one of the inverters in U-45, a 75LS04.

10. A jumper is also connected from pin 11 of U-45 to pin 3 of U-46. This connects "DBIN" to the input of the inverter in the previous item.

These are the only changes required to make the P.C. run, but a couple of additional items may be of interest to readers. The AM0026 specified as U-104 is hard to find and an MMH0026 works well. Note that this IC should be soldered in place. Also, don't use RCA CD 4029s for U-11 or U-1. I went through a half a dozen before substituting a Motorola MC-14029. The input protection or some other idiosyncrasy prevents the RCA chip from working in these circuits.

All in all, I now have a marvelously operating Sol for my trouble and a total of about $300.00 invested.

One other item may be of interest. I have designed a bi-directional parallel-to-serial conversion board for the Sol parallel port. This board allows Solos software to be used to select printers, terminals, etc. Any parallel port I/O is converted to or from serial form at any of eight switch selectable baud rates. Full handshaking is implemented and the board uses the Sol power supply. It is fully RS232C compatible and is totally software compatible with the Sol.

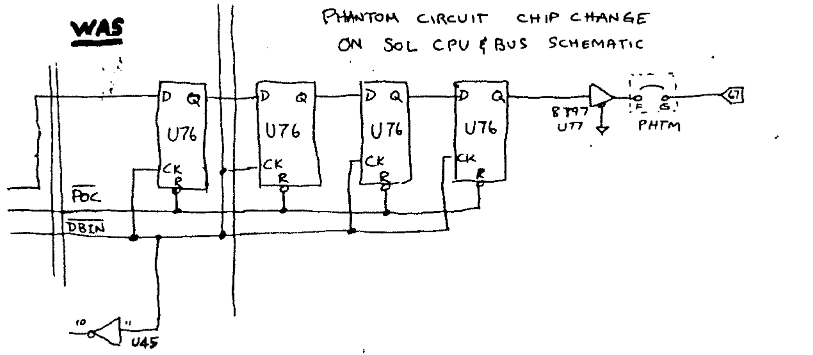
A CPM driver which selects the serial or parallel ports for "List" or "Console" devices is also available as a ready to run ".COM" file. It supports the Diablo and other printer's handshaking requirements.

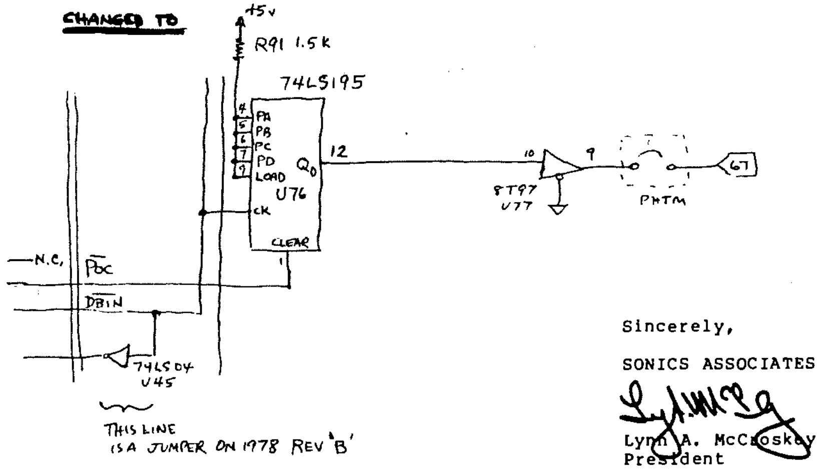
I use the board and CPM driver to connect a Soroc IQ-120 terminal as the console device while maintaining serial printer access.

The boards are available to Proteus members assembled and tested for $150.00 each. They come with complete instruction manuals. The CPM driver is available for $15.00 on eight inch single density format only. Anyone wishing to purchase either of these products should call me at 205/942-9631.

Thanks for a great newsletter. I hope this information will be helpful.

PHANTOM CIRCUIT CHIP CHANGE ON SOL CPU & BUS SCHEMATIC





Sincerely,

SONICS ASSOCIATES, INC.

Lynn A. McCroskey
President

```
*************************************************************
*************************************************************
****                                                     ****
****          1 6 K R A    D A T A    D E L A Y          ****
****                                                     ****
*************************************************************
*************************************************************
```

October 15, 1981

To whom it may concern:

The 16KRA DATA DELAY LINE reffered to by Joe Maguire & Ed Meyer in PROTEUS/NEWS, Vol. 4, No. 3/4, Pgs. 3 & 31, is a currently manufactured device. I have purchased two of these Digital Delay Units from the following source with 100% success in 3 different Processor Technology 16KRA's.

SOURCE:

DATA DELAY DEVICES
385 Lakeview Avenue
Clifton, N.J. 07011

Part No. DDU-4-7672
Phone: (201) 722-1106

PRICES:

No. of Units :	1-9	10-49	50-99	100+
Cost per Unit:	$20.00	17.00	13.00	10.00

SPECIFICATIONS: 14 pin DIP package:

```
                        ----------
             INPUT-->|1      14|---Vcc(+5V)
                    -|2      13|-
                    -|3      12|-->100 ns DELAY
  150 ns DELAY<--|4      11|-
                    -|5      10|-->250 ns DELAY
  350 ns DELAY<--|6       9|-
              GND---|7       8|-
                        ----------
```

Rise-time: 4 ns typical
Delay : +/-5% or 2 ns(whichever is greater)

PARAMETER		CURRENT	VOLTAGE	FAN-OUT
INPUT:	HIGH	50 uA max		
	LOW	-2 mA max		
OUTPUT:	HIGH		2.5 V min	20/tap max
	LOW		0.5 V max	10/tap max

POWER DISSIPATION: 19 mW/Gate

David Reis

David Reis
1843 Vassar Avenue
Mtn. View, CA 94043

HELP NEEDED

A request for help has come from Nat Pulsifer. He wants to contact someone that has ACC PAC ACCOUNTS/RECEIVABLE (Proteus Item # P5 and ACC PAC ACCOUNTS/PAYABLE (Proteus Item # P6) up and working.

Please contact:
Nathaniel Pulsifer & Associates
Investment Management & Financial Planning
Odd Fellows Professional Bldg.
Ipswich, MA 01938
(617) 356-3530

FOR SALE

LINE PRINTER: CENTRONICS MODEL 306 8.5" LINE LENGTH COMMERCIAL QUALITY, HEAVY DUTY, 100 CPS 40 LPM, NORMAL 7X9 DOT MATRIX, WITH EXPANDED CHARACTER CAPABILITY. EXCELLENT CONDITION VERY LIGHT USEAGE. WITH A WOODEN STAND, AND CASTERS FOR EASY MOVEMENT, PARALLEL INTERFACE, WORKS FINE WITH SOL SOLUS DRIVERS. $400.00 DELIVER IN THE BAY AREA, OTHERWISE FOB FREMONT, CALIF.

FRIDEN 7102 TTY RS 232 SERIAL INTERFACE. WORKS EXCELLENT WITH SOL 20 SOLUS, REQUIRES A FEW NULLS ON LONG LINES. TAPE READER AND TAPE PUNCH CAPABILTY, 20" CARRIAGE FOR EXTRA LONG LINES. HAS BEEN MAINTAINED AND IS IN VERY GOOD OPERATING CONDITION. PLUGS DIRECTLY TO SOL 20 SERIAL OUTPUT. 110 BAUD RATE APPROX 15 CPS. CASSETTE COPY OF NORTH STAR DOS IF BUYER NEEDS THE DRIVER. $150.00 DELIVER IN THE BAY AREA OTHERWISE FOB FREMONT, CALIF.

MILLARD MCKINNEY
43,337 ISLE ROYAL ST.
FREMONT, CA 94538
(415) 655-7393

FOR SALE

--Helios II System and a Sanyo monitor, plus Okidata 110 printer (RS232 and tractor). All low mileage. Includes SOL Rev. D with 48K memory, PTDOS 1.5(mod 2), serial port driver for printer and some games. All offers will be considered.
Rod Lee, 5 West Creek Court, Lafayette CA 94549, (415)836-9566.

HELP WANTED

As a 2 year subscriber to Proteus, I have a request for help. I'm trying to bring up UCSD Pascal on an S-100 system that has Processor Technology's Subsystem - B boards. The problem is in interfacing UCSD Pascal's setup and X-Y cursor addressing schemes with the VDM-1. All seems well except the UCSD Editor is not useable. Can any members of Proteus offer any help?

Thanks,
Kerry Montgomery
13420 S.W. Castlewood
Beaverton, Oregon 97005

WANTED

I need a copy of the CUTER OBJECT TAPE, please contact me if you can help.

Thomas Roman, 441 Amsterdam Ave., N.Y.,N.Y. 10024 (212)496-0442

TABLE OF CONTENTS

RENEWAL NOTICE

IT'S TIME TO RENEW YOUR SUBSCRIPTION TO PROTEUS FOR 1982.

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, $24. per year for U.S.A., CANADA, and MEXICO. $32. per year for other foreign addresses.

ALL SUBCRIPTIONS RECEIVED AFTER MARCH 31, 1982 WILL BE $30 PER YEAR FOR U.S. AND CANADA AND MEXICO, AND $38 PER YEAR OF OTHER FOREIGN ADDRESSES. THE EXTRA COST IS TO COVER REPRINTING OF BACK ISSUES FOR 1982 FOR LATE SUSCRIBER.

PROTEUS/NEWS will be published as a quarterly in 1982.

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 94061-3483,USA, telephone (415) 368-2300.

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

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

Joe Maguire
PO Box 3742 DT
Anchorage, AK

99510