

PROTEUS / NEWS

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

FORMERLY SOLUS NEWS

September/October 1981
November/December

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 eliminating 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 dismantled 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 noticeable, 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.

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.

Enclosed 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 Receivables, 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
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

Page 1

ACCT. NO.	ACCOUNT DESCRIPTION	DEBIT	CREDIT
100	Rental Income		\$30,539.52
101	Vending Income		589.91
102	Late Charges		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	Advertising	68.74	
208	Maintenance and Repairs	3,829.34	
209	Paintins and Decoratins	83°	
210	ABC Managemen!		
211	Resident "		

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 Vendins 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	
84.32	92.50	6.18	7	202 Gas & Electric	1035.84	1320.85	285.01	28	
83.75	65.00	-18.75	-22	203 Gardenins	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 & Garbase	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 Advertisins	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 Paintins and Decoratins	589.56	839.13	249.57		
116.25	125.00	8.75	8	210 ABC Manasement	1395.00	1711.07	316.07		
100.00	100.00	.00	0	211 Resident Manaser	1200.00	815.00	-385.00		
8.00	.00	-8.00	-100	212 Cleanins Equip		68.00	68.00		
		.00	0	213 Lesal Exp		.00	.00		
		25.93	-100	214 Misc					
		.00	-3	215 In					

TRANSACTION: EXPENSEX - CHECKS 187 TO 195

4	202	Gas & Electric	7	12/31/1981	92.50	
	203	Gardenins	8	12/31/1981	65.00	
	204	Pest Control	9	12/31/1981	7.50	
	205	Water, Sewer & Garbase	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	Paintins and Decoratins	13	12/31/1981	25.00	
	210	ABC Manasement	14	12/31/1981	125.00	
	211	Resident Manaser	15	12/31/1981	100.00	
	300	Cash & Checkins Accounts	16	12/31/1981		1039.98

TRANSACTION: MORTGAGE PAYMENT - FIRST SAVINGS & LOAN

5	300	Cash & Checkins 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 & Checkins 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 & Checkins 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
TOTAL ASSETS \$236,196.23 \$240,039.80

CURRENT LIABILITIES

Cleanins Deposits \$270.00 \$270.00

LONG TERM LIABILITIES

Mortgage Payable \$135,246.63 \$135,246.63
TOTAL LIABILITIES \$135,246.63 \$135,516.63

CAPITAL

Equity \$107,009.08
 PROFIT OR LOSS -2,485.91
TOTAL CAPITAL AND LIABILITIES \$104,523.17 \$240,039.80

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 & Checkins Accounts	2	12/31/1981	2475.00	

TRANSACTION: VENDING - AAA VENDING

2	101	Vendins Income	3	12/31/1981		54.63
	300	Cash & Checkins 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 & Checkins Accounts	6	12/31/1981		1256.37

PASA ROBLES APARTMENTS

PREPARED 12/31/1981 AS OF 12/31/1981

GENERAL LEDGER TRIAL BALANCE		Page 1				
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

EXPENSES

Property Taxes	\$1,256.37
Insurance	1,300.00
Gas & Electric	1,320.85
Gardening	863.00
Pest Control	90.00
Water, Sewer & Garbase	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

PASA ROBLES APARTMENTS

PREPARED 12/31/1981 AS OF 12/31/1981

GENERAL LEDGER TRIAL BALANCE		Page 2				
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 & Checkings 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				

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 sector 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 sector 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.

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

USER REACTS TO 24 X 80 CONVERSION
by Earl Dunham

Stans:

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 Hogg's 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 Stans:

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, Terrific, 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 incredible. 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 Pissy-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 already 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 Pissy-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 disassembled 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 characters 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 necessary 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 development 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 UOPRT
;
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 OUTPUTTING 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
BB2C JZ BB29
BB2F CPI 80
BB31 JNZ BB36
BB34 MVI A,03
BB36 CPI 8B
BB38 JZ BB49
BB3B CPI 8C
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
BB5F LDA BB54
BB62 CPI 7F
BB64 JNZ BB91
```

;Continue CONIN below

;No change from SOLUSER9

```
;This patch replaces Dr. Byrams delete routine
;This area is travelled twice for some
;deletions.
```

```
BB67 MVI A,00
BB69 STA BB54
BB6C MVI B,5F
BB6E JMP BB8E
BB71 CALL C02E
BB74 STA BB53
BB77 STA BB54
BB7A RET
BB7B NOP
BB7C NOP
BB7D NOP
BB7E NOP
BB7F NOP
BB80 NOP
BB81 MOV B,C
BB82 LDA BB55
BB85 CPI 0D
BB87 JNZ BB8E
BB8A CMP C
BB8B JZ BB91
BB8E CALL C054
BB91 POP B
BB92
```

```
;Call KSTAT
;Deletes are intercepted and stored.
;Watch out below a delete is coming through
Return to continue CONIN
```

;No changes

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 versatile; 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 manufacturers), 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 compatible 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 synchronization 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 compatibility. 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

```
*****
** 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
;taps, 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 OFFH ;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 OFFH ;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
```

```
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-FILED
```

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 W. 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
4	7	7	7	7	7	7	7	7	7	7	7
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 1FFFFH 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 AS="HELLO" \ PRINT FNR$(AS)
30 END
9000 DEF FNR$(XS)
9005 FOR I=1 TO LEN(XS) \ X=ASC(XS(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 \ XS=CHR$(27)+CHR$(k)+CHR$(X)
9010 PRINT XS, \ Y=CALL(49462) \ XS="" \ GOTO 9020
9015 XS=CHR$(27)+CHR$(1)+CHR$(X)+CHR$(27)+CHR$(2)+CHR$(Y)
9020 RETURN XS \ 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; OC80AH).

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
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 RENAME 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      0090 *   3. Add the new command table entry at 2141H by
2900      0095 *   entering the ASCII values of the command name.
2900      0100 *   i.e., "RN", followed by the jump address to RNAME
2900      0105 *   at 2900H. NOTE: There is also room for two more
2900      0110 *   command table entries if you wish to add them.
2900      0115 *   You may want to consider adding "BA" for a jump
2900      0120 *   to BASIC at 2A04H, and "BC" for a jump to 2A00H.
2900      0125 *   If additions are made, don't forget to repeat
2900      0130 *   step 2, above. (Also, don't forget that all
2900      0135 *   addresses are entered into memory low byte first.)
2900      0140 *
2900      0145 * NOTE: Do not make these changes unless your DOS is rel.4
2900      0150 *
2900 CD A5 23      0155 DEL   CALL 23A5H
2903 CD 4D 27      0160      CALL 274DH
2906 CD FA 23      0165      CALL 23FAH
2909 C3 04 24      0170      JMP   2404H
290C      0175 *
290C CD A5 23      0180 RNAME  CALL 23A5H  RENAME EXTENSION
290F E5            0185      PUSH H
2910 3A F0 28      0190      LDA  28F0H  UNIT#
2913 F5            0195      PUSH  6
2914 CD B7 23      0200      CALL 23B7H
2917 F1            0205      POP   6
2918 32 F0 28      0210      STA  28F0H
291B C3 BF 22      0215      JMP   22BFH
291E      0220 *
291E      0225 * The SPOOLER employs the memory from 0000H to 1FFFH as
291E      0230 * a circular buffer for output to a printer. Therefore,
291E      0235 * it is assumed that memory is available in this area and
291E      0240 * that there is no conflicting memory use.
291E      0245 *
291E      0250 * The DOS JUMP TABLE from 2000H to 2019H will, of course,
291E      0255 * have to be matched to reflect the new routine addresses:
291E      0260 *      200D   JMP   OUTPB
291E      0265 *      2010   JMP   KEYIN
291E      0270 *      2013   JMP   TINT
291E      0275 *      2016   JMP   CTRLC
291E      0280 *
291E C3 D5 C0      0285 TINT  JMP   0C00SH  ERASE SCREEN (For std. SOLOS only.)
2921      0290 *
2921 00 00          0295 INPTR  DW   00H   SPOOLER INPUT POINTER
2923 00 00          0300 OTPTR  DW   00H   SPOOLER OUTPUT POINTER
2925 00          0305 WAITF  DB   00H   PRINTER BUSY FLAG

2926 00          0310 INPF  DB   00H   INPUT FLAG
2927 00          0315 OUTF  DB   00H   OUTPUT FLAG
2928      0320 *
2928 E6 03          0325 KEYIN  ANI  03H
292A 32 26 29      0330      STA  INPF
292D CD A2 29      0335 WAIT1  CALL  SPOUT
2930 3A 26 29      0340      LDA  INPF
2933 CD 22 C0      0345      CALL  AINP  0C022H  SOLOS AINP RTN.
2936 CA 2D 29      0350      JZ   WAIT1
2939 FE 80          0355      CPI  80H
293B CA 04 C0      0360      JZ   0C004H  OPTIONAL JUMP TO SOLOS ON MODE SELECT
293E FE 03          0365      CPI  03H  CONTROL-C?
2940 CC 6A 29      0370      CZ  STOP  KILL PRINTER OPERATION
2943 E6 7F          0375 CNT1  ANI  7FH
2945 FE 7F          0380      CPI  7FH  61H FOR UPPER CASE ONLY
2947 D8            0385      RC
2948 D6 20          0390      SUI  20H
294A C9            0395      RET
294B      0400 *
294B E6 03          0405 OUTPB  ANI  03H
294D C2 55 29      0410      JNZ  CNT2
2950 CD 19 C0      0415      CALL  0C019H  SOLOS SOUT RTN.
2953 78            0420      MOV  A,B
2954 C9            0425      RET

```

```

2955 FE 01          0430 CNT2  CPI  01H  CHECK FOR PRINTER PSEUDO PORT
2957 CA 73 29      0435      JZ   SPINP  SPOOL IT IF PRINTER PORT SELECTED
295A CD 1C C0      0440      CALL  AOUT  OTHERWISE, USE 0C01CH SOLOS AOUT RTN.
295D 78            0445      MOV  A,B
295E C9            0450      RET
295F      0455 *
295F CD A2 29      0460 CTRLC  CALL  SPOUT
2962 CD 1F C0      0465      CALL  SIMP  0C01FH SOLOS SIMP RTN.
2965 E6 7F          0470      ANI  7FH
2967 FE 03          0475      CPI  03H
2969 C0            0480      RNZ
296A E5            0485 STOP  PUSH  H
296B 2A 21 29      0490      LHL  INPTR
296E 22 23 29      0495      SHLD  OTPTR
2971 E1            0500      POP  H
2972 C9            0505      RET
2973      0510 *
2973 C5            0515 SPINP  PUSH  B   SPOOLER BUFFER INPUT
2974 D5            0520      PUSH  D
2975 E5            0525      PUSH  H
2976 2A 21 29      0530      LHL  INPTR
2979 70            0535      MOV  A,B
297A CD E8 29      0540      CALL  INCPR
297D 22 21 29      0545      SHLD  INPTR
2980 CD E8 29      0550      CALL  INCPR
2983 EB            0555      XCHG
2984 2A 23 29      0560      LHL  OTPTR
2987 CD E9 29      0565      CALL  INCPR+1
298A C2 D6 29      0570      JNZ  EXIT
298D CD D8 29      0575 WAIT2  CALL  PBUSY
2990 C2 8D 29      0580      JNZ  WAIT2
2993 AF            0585      XRA  A
2994 32 25 29      0590      STA  WAITF
2997 4F            0595      MOV  C,A
2998 CD A2 29      0600 DT256  CALL  SPOUT
299B 0C            0605      INR  C
299C C2 98 29      0610      JNZ  DT256
299F C3 D6 29      0615      JMP  EXIT
29A2      0620 *

29A2 CD D8 29      0625 SPOUT  CALL  PBUSY  SPOOLER BUFFER OUTPUT
29A5 C0            0630      RNZ
29A6 AF            0635      XRA  A
29A7 32 25 29      0440      STA  WAITF
29A8 C5            0645      PUSH  B
29A9 D5            0650      PUSH  D
29AC E5            0655      PUSH  H
29AD 2A 21 29      0660      LHL  INPTR
29B0 EB            0665      XCHG
29B1 2A 23 29      0670      LHL  OTPTR
29B4 CD E9 29      0675      CALL  INCPR+1
29B7 CA D6 29      0680      JZ   EXIT
29BA 46            0685      MOV  B,M
29BB 3E 01          0690      MVI  A,01H  SELECT OUTPUT DEVICE
29BD CD 1C C0      0695      CALL  AOUT  0C01CH SOLOS AOUT ROUTINE
29C0 CD E8 29      0700      CALL  INCPR
29C3 22 23 29      0705      SHLD  OTPTR
29C6 78            0710      MOV  A,B
29C7 FE 00          0715      CPI  00H
29C9 C2 D6 29      0720      JNZ  EXIT  This instruction is for ETX/ACK protocol
29CC      0725 *   printers (QUE, DIABLO, NEC, etc.) only.
29CC      0730 *   The instruction should be "JMP EXIT" for
29CC      0735 *   printers which employ CLEAR-TO-SEND or
29CC      0740 *   DATA-SET-READY protocols (TI-810, ANADIX
29CC      0745 *   PRINTERM, BASE II, etc.).
29CC      0750 *
29CC 3E 01          0755      MVI  A,01H  SELECT OUTPUT PORT
29CE 32 25 29      0760      STA  WAITF
29D1 06 03          0765      MVI  B,03H

```

CON'T FROM PAGE 13-PRINT SPOOLING FOR N*

```

2903 CD 1C C0 0770 CALL AOUT (OC01CH) SEND ETX TO PRINTER
2906 E1 0775 EXIT POP H
2907 D1 0780 POP D
2908 C1 0785 POP B
2909 78 0790 MOV A,B
290A C9 0795 RET
290B 0800 *
290C 0805 * The routine following is for ETX/ACK printers. If the
290D 0810 * printer you have uses DATA-SET-READY or CLEAR-TO-SEND,
290E 0815 * substitute the routine below in place of the other:
290F 0820 *
2910 0825 * PBUSY IN OFBH (Get serial status)
2911 0830 * ANI 02H (20H for CLEAR-TO-SEND printers)
2912 0835 * RET
2913 0840 *
2914 0845 PBUSY LDA WAITF PRINTER BUSY?
2915 0850 ORA A
2916 0855 RZ
2917 0860 MVI A,01H SELECT INPUT DEVICE
2918 0865 CALL AIMP OC022H SOLOS AIMP RTN.
2919 0870 CPI 06H ACK RECEIVED?
2920 0875 RET
2921 0880 *
2922 0885 INCR INX H INCREMENT POINTER,
2923 0890 MOV A,H AND COMPARE HL & DE
2924 0895 ANI 1FH MODULUS 2000
2925 0900 MOV H,A
2926 0905 CPH D
2927 0910 RNC
2928 0915 MOV A,L
2929 0920 CPH E
2930 0925 RET
2931 0930 *
2932 0935 * END OF PROGRAM *

```

```

2900 0005 * PRINT VIDEO SCREEN ROUTINE FOR SOL-20
2901 0010 * (Using North Star or CP/M DOS systems)
2902 0015 *
2903 0020 * (C)1980 ARMSTRONG CORPORATION
2904 0025 * P.O. Box 10533
2905 0030 * Costa Mesa, CA 92627
2906 0035 *
2907 0040 * To integrate the print screen routine into your DOS,
2908 0045 * provide a jump to SCNPT in an appropriate place in
2909 0050 * your Console input routines. For example:
2910 0055 *
2911 0060 INPUT ANI 03 This is the North Star version.
2912 0065 STA FLAG Save input device number.
2913 0070 INPI LDA FLAG
2914 0075 CALL AIMP SOLOS AIMP ROUTINE (OC022H)
2915 0080 JZ INPI Loop until character received.
2916 0085 CPI 140 SOL-20 LOAD KEY PRESSED?
2917 0090 JZ SCNPT If yes, then jump to screen print routine.
2918 0095 * (Regular program flow continues here)
2919 0100 *
2920 0105 FLAG DB 0
2921 0110 EQU * FOR DEMO PURPOSES ONLY
2922 0115 EQU *
2923 0120 EQU *
2924 0125 EQU *
2925 0130 EQU *
2926 0135 EQU *
2927 0140 EQU *
2928 0145 EQU *
2929 0150 EQU *
2930 0155 EQU *
2931 0160 EQU *
2932 0165 EQU *
2933 0170 EQU *
2934 0175 EQU *
2935 0180 EQU *
2936 0185 EQU *
2937 0190 EQU *
2938 0195 EQU *
2939 0200 EQU *
2940 0205 EQU *
2941 0210 EQU *
2942 0215 EQU *
2943 0220 EQU *
2944 0225 EQU *
2945 0230 EQU *
2946 0235 EQU *
2947 0240 EQU *
2948 0245 EQU *
2949 0250 EQU *
2950 0255 EQU *
2951 0260 EQU *
2952 0265 EQU *
2953 0270 EQU *
2954 0275 EQU *
2955 0280 EQU *
2956 0285 EQU *
2957 0290 EQU *
2958 0295 EQU *
2959 0300 EQU *
2960 0305 EQU *
2961 0310 EQU *
2962 0315 EQU *
2963 0320 EQU *
2964 0325 EQU *
2965 0330 EQU *
2966 0335 EQU *
2967 0340 EQU *
2968 0345 EQU *
2969 0350 EQU *
2970 0355 EQU *
2971 0360 EQU *
2972 0365 EQU *
2973 0370 EQU *
2974 0375 EQU *
2975 0380 EQU *
2976 0385 EQU *
2977 0390 EQU *
2978 0395 EQU *
2979 0400 EQU *
2980 0405 EQU *
2981 0410 EQU *
2982 0415 EQU *
2983 0420 EQU *
2984 0425 EQU *
2985 0430 EQU *
2986 0435 EQU *
2987 0440 EQU *
2988 0445 EQU *
2989 0450 EQU *
2990 0455 EQU *
2991 0460 EQU *
2992 0465 EQU *
2993 0470 EQU *
2994 0475 EQU *
2995 0480 EQU *
2996 0485 EQU *
2997 0490 EQU *
2998 0495 EQU *
2999 0500 EQU *

```

```

2914 0125 NCHAR EQU OC80BH
2915 0130 BOT EQU OC80AH
2916 0135 *
2917 0140 SCNPT PUSH H SAVE MOST REGISTERS
2918 0145 PUSH D
2919 0150 PUSH B
2920 0155 CALL CREM REMOVE CURSOR
2921 0160 LDA BOT A HAS SCROLL OFFSET
2922 0165 INR A
2923 0170 LXI H,OC800H HL HAS NORMAL START OF SCREEN
2924 0175 LXI D,64 DE HAS ONE LINE DIFFERENCE
2925 0180 SCNI DAD D ADD ONE LINE
2926 0185 DCR A FOR EACH LINE OFFSET
2927 0190 JNZ SCNI LOOP UNTIL DONE
2928 0195 MOV C,A CLEAR REG. C
2929 0200 LXI D,1024 DE GETS SCREEN COUNTER
2930 0205 SCNZ MOV A,C C KEEPS #-CHARACTERS PER LINE
2931 0210 ANI 63 NO MORE THAN 64 FOR SOL-20
2932 0215 MOV C,A SAVE IT
2933 0220 JNZ SCNZ ONWARD IF NOT 64
2934 0225 MVI B,13 CARRIAGE RETURN
2935 0230 MVI A,1 FOR PRINTER
2936 0235 CALL OUTPB GOES OUT TO PRINTER
2937 0240 MVI B,10 FOLLOWED BY LINE FEED
2938 0245 MVI A,1
2939 0250 CALL OUTPB
2940 0255 SCNZ MOV A,H NOW REGULAR BUSINESS
2941 0260 ANI 7FH GET CHARACTER & STRIP PARITY
2942 0265 CPI 32 TEST FOR CONTROL CHARACTER
2943 0270 JNC SCNA ONWARD IF OKAY
2944 0275 MVI A,63 OTHERWISE IT'S A "?"
2945 0280 SCNA MOV B,A SAVE IT IN B
2946 0285 MVI A,1 SET FOR PRINTER OUTPUT
2947 0290 CALL OUTPB AND SEND IT
2948 0295 CALL CNTLC CHECK FOR CONTROL-C
2949 0300 JZ SCNB ONWARD IF SO
2950 0305 INX H POINT TO NEXT CHARACTER
2951 0310 INR C AND BUMP LINE COUNTER
2952 0315 MOV A,H CHECK FOR WRAP AROUND
2953 0320 CPI 000H ARE WE THERE?
2954 0325 JNZ SCNC ONWARD IF NOT
2955 0330 MVI H,OC0CH OTHERWISE, WRAP AROUND
2956 0335 SCNC DCX D ONE LESS CHARACTER TO WORRY US
2957 0340 MOV A,D SINCE DCX SETS NO FLAGS,
2958 0345 ORA E THIS IS HOW WE CHECK FOR ZERO IN DE
2959 0350 JNZ SCNZ MORE WORK AHEAD
2960 0355 SCNB CALL VDADD PUT CURSOR BACK
2961 0360 MOV A,H
2962 0365 ORI 80H
2963 0370 MOV M,A
2964 0375 POP B RESTORE ALL REGISTERS
2965 0380 POP D
2966 0385 POP H
2967 0390 JHP INPUT PRETEND WE'RE STILL WAITING FOR KEYINP
2968 0395 *
2969 0400 * NOTE: For CP/M systems, lines 0235, 0250, 0290 and
2970 0401 * 0295, above, will have to be modified to reflect
2971 0405 * the differences in the output routines of the two
2972 0410 * operating systems. One method might look like this:
2973 0411 *
2974 0415 * OUTPB PUSH H
2975 0420 * PUSH D
2976 0425 * PUSH B
2977 0430 * MOV C,B
2978 0435 * CALL LIST
2979 0440 * POP B
2980 0445 * POP D
2981 0450 * POP H
2982 0455 *
2983 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 matters 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 recently 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 Htype 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 shiny 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!

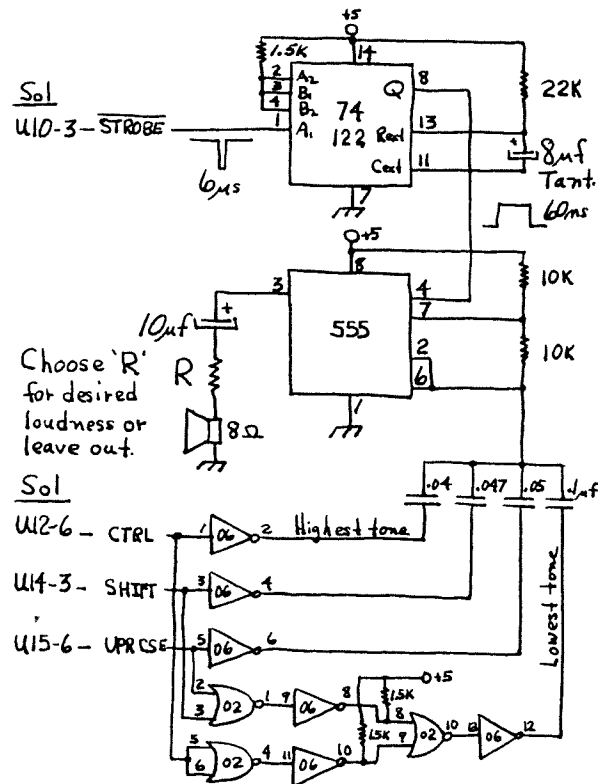
Adding tones to Sol's keyboard is not a new trick. I've seen many articles on it, but included is a schematic for one I did that may be of interest to some as it gives different tone levels to the 4 main keyboard outputs. These are unshifted, shifted (shift lock), alpha shift (upper case) and control. 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 (7415138) 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, SFT1 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 continuing 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. 1980 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 30K 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 30K 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 incompatibility between the two systems. To write a file onto a SOLOS tape, I use DDT to load the file into memory, then execute a GC004 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 0100              (end address, start address)
-GC004                (go to SOLOS)
>SET TY=43             (in SOLOS, set file type to C)
>SAVE EDIT 100 13FF    (save on tape from 100 to 13FF)
>EX 0                  (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 0100 12FF      (SOLOS reads the file in)
>EX 0                 (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 #1 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, 2307 Center St. Kingsport, Tn. 37660. 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 occurred 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
Don L. Finley
310 Willow St.
Mt. Carmel, TN.
37642
(615) 357-3355

```
*****
; PTDOS TO CPM FILE TRANSFER
; Don L. Finley 10-22-80
;
; 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 PRECEDED 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 0
; 2. UPPERCASE-REPEAT
; 3. REMOVE DISK AND REPLACE WITH A 30K CPM SYSTEM DISK
; 4. BOOT ON 30K CPM IN DRIVE 0
; 5. PLACE PTDOS DISK IN DRIVE 1
; 5. TYPE DOWNLOAD CPMNAME.EXT PTNAME
; 6. SET BACK AND WAIT TILL FINISHED
*****
ORG 100H
LXI SP,STACK
LXI D,MSGL ;PRINT SIGN-ON MESSAGE
MVI C,PSTRIN
CALL BDOS
CALL ASKOK ;SEE IF USER WANTS TO CONTINUE
JNZ 0 ;IF NOT, THEN A RETURN TO CPM
LDA 80H ;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 '/' (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
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 RUFFER 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,09H      ;PUT IN THE TAB
;        CALL PUTC      ;PUT IN CPM'S OUTPUT
; SPCOI: CALL GETCH     ;GET NEXT CHARACTER
;        CPI ' '        ;KEEP LOOPING TILL NO MORE SPACES
;        JZ SPCOI
;        JMP PROCS1     ;BACK TO NORMAL PROCESSING
;
; ; PROCESSING FOR END OF LINE (CARRIAGE-RETURN)
; ; NEED TO SEND CR AND ADD A LINE-FEED
;
; EOL:   CALL PUTC      ;PUT IN CARRIAGE RETURN
;        MVI A,LF      ;AND THEN INSERT A LINE-FEED
;        CALL PUTC     ;
;        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 PUTC      ;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 80H + 128)=ADDRESS 100H
; ; IF THE BUFFER IS FULL IT IS WRITTEN TO THE CPM FILE ON
; ; DRIVE 0.
;
; PUTC:  MOV B,A        ;SAVE CHARACTER IN THE B-REG
;        LHLD CPMAD    ;GET ADDRESS OF CPM DATA
;        MOV A,H       ;SEE IF REACHED ADDRESS 100H 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 0).
; ; 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,80H  ;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,200H    ;READ IN 200H 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,200H   ;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 100H ADDRESS
;        CPI 1
;        JZ ALLD1    ;WHEN THERE, ALLL 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
;
PTERR1: LXI D,MSG3 ;PRINT MESSAGE STATING PTDOS
        MVI C,PSTRIN ;FILE NOT FOUND
        CALL BDOS
        JMP QUIT ;QUIT IF PTDOS FILE DOESN'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,'S'
ERRMS1: DB 'PT ERROR # $'
ERMS1: DB 'EXPECTING FILE NAME',CR,LF,'S'
ERMS2: DB 'INVALID PTDOS FILE NAME',CR,LF,'S'
ERMS3: DB 'DISK FULL!!!',CR,LF,'S'
MSG3: DB 'PTDOS FILE DOES NOT EXIST',CR,LF,'S'
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
```

```
ERR1: DB 'HIT RETURN TO CONTINUE $'
      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 programs do not use the 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

```
TITLE 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 BJORN DAL'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(0) 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 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 74504 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 DTLDM-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 propagation 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 Expandram (4115 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 SOLDS 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 regret 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 resemblance 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 quantities of 100. Smaller quantities 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
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	DFORT
		ANI	07FH
		RET	
		DS	16
CHANGE TO	CINP	IN	STATP
		CMA	
		ANI	IRDY
		JZ	CINP
		IN	DFORT
		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?

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 "bbb-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

further 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/VDML 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;
        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
  minusign = '-';
  plusign = '+';
  ascizero = 48;
var
  ch : char;
  ptr, ncl, ncr, dpos : integer;
  realval,mult : real;
  minus : boolean;

procedure decodeeach;
begin

```

```

  realval := realval
    + ((ord(s.val[ptr]) - ascizero) * mult);
end;{decodeeach}

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-numeric}

    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

```

```

end;

  { Convert characters left of decimal point }

  mult := 1;
  for ptr := ncl downto 1 do
    begin
      decodeeach;

      mult := mult * 10
    end;

  { Convert characters right of decimal point }

  mult := 0.1; ptr := dpos + 1;
  while ncr > 0 do
    begin
      decodeeach;
      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, CA 91320 }



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.

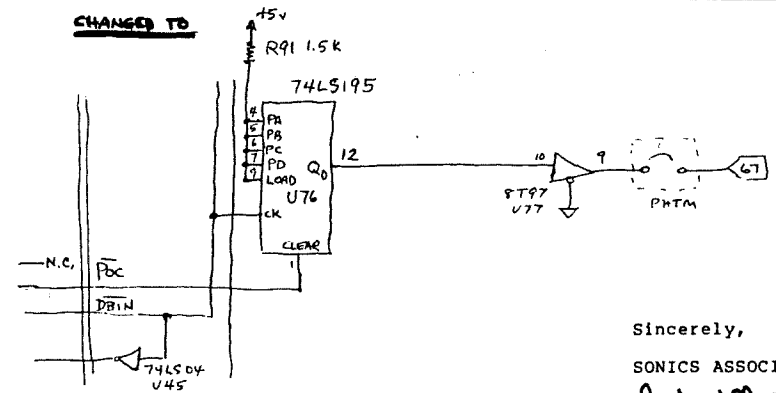
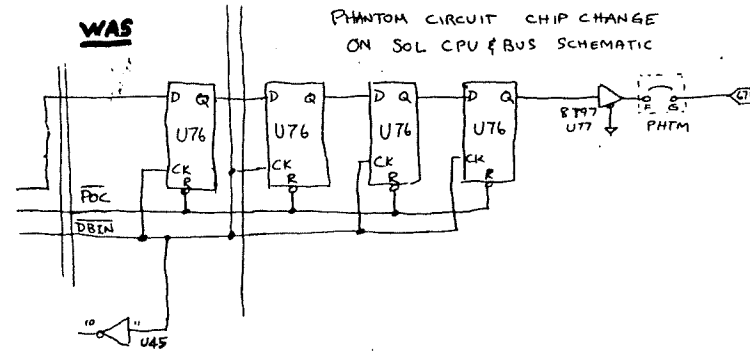
One other item may be of interest. I have designed a bi-directional parallel-to-serial conversion board for the Sol parallel port. This board allows Solos software to be used to select printers, terminals, etc. Any parallel port I/O is converted to or from serial form at any of eight switch selectable 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.



THIS LINE IS A JUMPER ON 1978 REV 'B'

Sincerely,
SONICS ASSOCIATES, INC.

Lynn A. McCroskey
Lynn A. McCroskey
President

16KRA DATA DELAY

October 15, 1981

To whom it may concern:

The 16KRA DATA DELAY LINE referred to by Joe Masuire & 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 Part No. DDU-4-7672
 385 LaKeview Avenue
 Clifton, N.J. 07011 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-->	11	14	---Vcc(+5V)
	12	13	---
	13	12	-->100 ns DELAY
150 ns DELAY<--	14	11	---
	15	10	-->250 ns DELAY
350 ns DELAY<--	16	9	---
	GND---	17	8

Rise-time: 4 ns typical
 Delay : +/-5% or 2 ns (whichever is greater)

PARAMETER	CURRENT	VOLTAGE	FAN-OUT
INPUT: HIGH	150 uA max		
LOW	-2 mA max		
OUTPUT: HIGH		12.5 V min	20/10p max
LOW		10.5 V max	10/10p 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 CAPABILITY, 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

UPGRADE SOL TO 24 X 80 SCREEN/Stan Sokolow.....	1
PUBLIC DOMAIN GENERAL LEDGER FOR N*/F.J. Hirner.....	3
HELIOS/MORROW DISK MULTIPLEXER ALLOWS HARD/SOFT SECTORING....	5
USER REACTS TO 24 X 80 CONVERSION/Earl Dunham.....	6
A RESPONSE TO D.PATTEN'S OPEN LETTER/A. T. Fincher.....	7
STILL MORE ON THE CPM USER AREA/D. Moller.....	8
VEIT IS HIGH ON SOL AND PROTEUS.....	8
WHO SAYS "CASSETTE TAPE IS DEAD"?/L. Moseley.....	9
SOL VANISHING TRICK-IT ALMOST WORKS.....	11
SOLUTION TO THE FILE DRIVER FOR NEC SPINWRITER/M.K. Gauthier.....	11
CORRECTION ON EPSON MX-80.....	11
PRINT SPOOLING FOR N*/W. D. Armstrong.....	12
SOL KEYBOARD TONE CIRCUIT-WITH 4 TONE LEVELS/H. Leon Winter.....	15
PTDOS TO CP/M FILE TRANSFER/D.L. Finley.....	16
GAMEPAC/1 DOESN'T RE-ACT/J.E. Breden.....	19
SLAC PASCAL PATCHES & BENCHMARKS/M. McKelvey.....	20
REGARDING R. BJORNNDAL'S RANDOMIZE FOR PT BASIC/M. Mackensie.....	21
MY DELAY LINE FAILED/R.A. Ellingsworth.....	22
WHAT'S NEW/ECBASIC SOURCE CODE REALLY ISN'T.....	22
VULCAN DATABASE MANAGER GIVES BIRTH TO dBASE II.....	22
WHERE TO FIND DISKS & RIBBONS/Mike McKelvey.....	23
FIX FOR VISTA CBIOS/L. Kalish.....	23
IS CP/M 2.2 FOR HELIOS OUT THERE?.....	23
ATOR:A PASCAL CONVERSION ROUTINE/D.S. Hunt.....	24
BUILDING THE 1978 REV B (NEW) SOL-PC/L.A. McCroskey.....	26
16 KRA DATA DELAY/D.Reis.....	27
UNCLASSIFIED ADS.....	27

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 SUBSCRIPTIONS 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.

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.

Copyright (C) 1981 by Proteus. All rights reserved. Permission is hereby granted to reproduce any computer programs contained herein, provided that Proteus and the program's author are given credit.

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

Joe Maguire
PO Box 3742 DT
Anchorage, AK

99510

