

1984 Free Disk Offer

Every 1984 member who registers before 1 June 1984 will receive a free disk. At the time of publication INSUA plans to send each newly registered member a floppy disk containing the DOS version of the editing program SECRETARY on one side and a series of CP/M MODEM712 programs on the other. Please allow six to eight weeks for delivery.

INSUA understands that the original publishers of SECRETARY have announced that this product is now in the public domain. INSUA has acquired a copy of the DOS version, and plans to place it on the DOS side of the free disk. (In the future, INSUA may be making SECRETARY available as a CP/M disk offering also, though as part of the disk library rather than as a free offering. Please make no inquiries before an announcement is made by INSUA.)

INSUA's MODEM712 offering consists of an update of the MODEM7 offered on INSUA disk #1013. MODEM712 has many features not available in previous versions of this famous Ward Christensen public domain communications program. The free INSUA disk will contain the following versions of MODEM712: Generic, Advantage Serial, Horizon HSIO4, Horizon standard serial, Micromodem 100, and PMMI 103. The serial port versions will support virtually any external RS-232 modem including the D. C. Hayes Smartmodem, with its main features. Most INSUA members should find a version they can simply boot up and run, on the condition that they have the proper modem hardware. At a minimum this hardware might consist of an external 300 baud modem (available for around \$100-150), plus a three-wire RS-232 cable (pin 7 to pin 7, 2 to 3, and 3 to 2) from the second serial port to the modem.

INSUA NEWS INSUA NEWS INSUA NEWS INSUA NEWS INSUA NEWS INSUA NEWS

Annual Meeting

The fifth annual meeting of INSUA will take place during the West Coast Computer Faire at the Civic Auditorium in San Francisco on Saturday, 24 March, 1984. The meeting will convene in Room 324 of the main Auditorium building at 1:30 p.m. The tentative schedule is as follows:

#

- 1:30 Welcome by the Chairman
- 1:40 Official balloting for 1984 board, conducted by Bill Banaghan
- 2:00 Report on the North Star Dimension (or other North Star developments) by a representative from North Star
- 2:40 Report on APCBASIC and the "Blue Binder" North Star software program by John Cleckner of the American Planning Corporation
- 3:20 Official meeting, followed by open discussion
- 3:45 Announcement of Board election
- 4:00 Meeting promptly adjourned

#

In Store for 1984

With the current issue of Compass, INSUA is making changes which we hope will improve its usefulness to members, without offending loyal traditionalists.

We have moved to a new format for Compass, a folded and saddle-stitched magazine which should be more uniform from issue to issue but will retain an approximately 8½" x 11" format and will continue to be punched for insertion into a three-hole binder. It is our plan to print six 32-page issues per year. Making this change has required the addition of a new paid employee to the INSUA staff; we welcome on board Valerie Kuletz, who is serving as editorial assistant!

In addition to the six regular issues, INSUA plans to print a **Special Issue** within the next two months. This **Special Issue** will be distributed free to all registered North Star owners as well as to all members of INSUA. Features of the **Special Issue** will include an **Index** to volumes I-III, and an updated Library report.

#

The Compass is published every two months by INSUA, the Interational North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, 2600 Buena Vista Way, Berkeley, CA 94708.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

- 2 Covill D. **XMODEM Transfer Protocol**
Description of standard MODEM7 protocol
- 5 Maguire J. **The Phantom Signal**
History and applications of phantom line--pin 67
- 7 Nelson A. **Dummy Lines**
Technique for forcing the printing of a "blank" line
- 8 Porter R. **Commentary**
Dumping BASIC variables and installing quad drives
- 9 Hogan S. **Computer Pollution Solution**
Fitting a fan filter to the Horizon
- 13 Lambert W. **Alien Equipment**
Ted Carnevale reviews alien S-100 boards
- 15 Lambert W. **WordStar Block Letters**
Creating large block letters w/ WS and MailMerge
- 16 **Portable Advantage**
Puget Sound Computer Service creates a portable
- 17 Maguire J. **Beyond 64K**
Principles of bank-switching
- 19 Maguire J. **Three Laws of Bank Switching**
Clip Tip
- 19 Morgenstern L. **MEMTODAT**
BASIC program to create DATA statements
- 21 **Vendors' Column**
Printer Buffer, N*CAT, AUTOTYPE editor
- 23 Appleby D. **Appleby on TurboDOS**
First impressions of TurboDOS
- 24 **Advantage Graphics**
List of vendors
- 25 Nelson A. **Ribbonizer Revisited**
Followup of earlier review (3.1 pp 29-30)
- 26 Swenson L. **Quic-n-easy**
Short review of program to assist in writing programs
- 27 Levy S. **User's Visit to North ***
1983 visit--impressions and opinions
- 28 Noll S. **Commentary**
INSUA policy, APCBASIC, and Advantage graphics
- 29 **MagicBind FollowUp**
Readers' reactions to MagicBind and Diablo 630
- 30 Brodie III R. **More on APC BASIC**
Portability of BASICS including N* and APC
- 31 Llorens A. **Whither Micro Design**
Problems with a promising disk controller
- 32 Maguire J. **Correction**
Correction to "Two-Column" article (3.4)
- 32 Johnson R. **Morrow Hard Disk**
Installing the 16Mbyte drive

XMODEM Transfer Protocol

By D.L. Covill

This article comes to us from the San Diego Computer Society via Foghorn, November 1983, by permission of the author. We cannot be certain that it corresponds precisely to any MODEM7 version offered by INSUA. -- Ed.

The file transfer protocol used in the MODEMx series of programs is a very good, though not perfect, protocol that has become something of a de facto standard, because it is in the public domain and has been widely distributed on various remote CP/M systems across the country.

The protocol was originally developed by Ward Christensen, who wrote the original MODEM.ASM. It has since been enhanced by several others. This writeup is based on an article by Kelly Smith in CP/M-Net News, January 81, extended to include discussion of CRC and batch transfers.

What is a protocol and why do we need one?

A protocol is, quite simply, the ritual courtesies exchanged by two separate but equal computer systems in order to insure an orderly and correct transfer of information. For short, interactive communications we can get along without one because the information transferred is visible, and if it looks wrong or we lose it we can immediately ask for it again.

The situation is different when transferring data or program files. The volume to be transferred is greater, thus there's a greater chance of transmission errors. It's not easy to tell whether an error has occurred (in COM files, for instance), and retransmitting the entire file would take too long and might only make things worse.

A transmission protocol provides:

- a. Positive control over the process and procedure.
- b. A means of detecting errors in the transmission process.

- c. The ability to retransmit any portion that has errors.

Note, however, that BOTH parties to the transmission must use the SAME protocol, or the ritual doesn't work. That's the reason for this document--so that people who want to (or have to) use a program other than MODEM7 can implement the XMODEM protocol and join the great army of public-domain file transmitters.

Control Characters Used

The following ASCII standard control characters are used in the protocol:

SOH Start of Heading	01H Ctrl-A
ACK Acknowledge	06H Ctrl-F
NAK Negative Acknowledge	15H Ctrl-U
EOT End of transmission	04H Ctrl-D
CAN Cancel	18H Ctrl-X

General Transmission Scheme

Data is sent in 128-byte numbered blocks, with a single **checksum** appended to each block. The receiving computer performs its own **checksum** as it acquires the incoming data, and upon completing each block compares its result with the **checksum** from the sending computer. If they match, it returns an ACK to the sender, meaning "received OK, send some more." If they don't match, a NAK is returned, meaning "that didn't look right, please send it again." This process continues until the entire file has been transmitted (or the number of errors causes one of the parties to give up).

Received data is stored in memory, then written to disk every 16 blocks.

Block Format

The sending computer transmits a block in the following form:

SOH (Start of Heading)	01H
Block#	8 bits
Complement of block#	8 bits
-128 data bytes-	8 bits each
Checksum	8 bits

The **checksum** is calculated by summing the SOH, the block number, the block number complement, and the 128 data bytes.

Starting the Transmission

The sending and receiving systems have to get "in sync" to start the transmission. This turns out to be very simple--the sending computer simply waits for an initial NAK from the receiving computer. He's sure to get one, because the receiving computer will "time out" looking for data and send the NAK as a signal that he didn't receive a data block. The sending computer knows this and uses it as a signal to start the transmission:

<u>Sending Computer</u>	<u>Receiving Computer</u>
(waiting)	(waiting)
	(times out)
	NAK (15H)
Data Block 1	
	ACK (06H)
Data block 2	
	ACK (06H)
...	

Re-transmitting a Block

What happens if the block is NAK'ed? Easy, the sending computer just re-sends the previous block.

<u>Sending Computer</u>	<u>Receiving Computer</u>
	...
Block #2 (errors)	NAK
Block #2 (again)	
	ACK
Block #3	
	...

But what if the sending computer never receives the ACK (or NAK)? The sending computer times out after 10 seconds, decides that it has failed, and re-transmits the block. This is the reason for block numbers--the receiver detects that this is the previous block all over again, throws it away, and returns an ACK, thereby catching up. The integrity of the block number is verified by summing the SOH (01 Hex) with the block number plus the complement of the block number--the result must be zero for a proper transfer (e.g., for block 7, 01+07+F8 = 00).

The sequence of events, then, looks like this:

<u>Sending Computer</u>	<u>Receiving Computer</u>
	...
Block #2	ACK (lost in transit)
Block #2 (again)	(Discards duplicate)
	ACK
Block #3	
	...

Concluding a Transmission

Normal completion of a data transfer concludes with an EOT (END OF TRANSMISSION, 04H) from the sending computer, with a final ACK from the receiving computer.

<u>Sending Computer</u>	<u>Receiving Computer</u>
-data block-	
ACK	EOT (04H)
	ACK
-end program-	-end program-

MODEM7 Batch Transfers

The MODEM7 program allows "Batch" file transfers--the sending program says something like

SB NEWSTUFF.*

and the receiving program says simply "RB".

The B sub-option means (a) that more than one file will be transmitted and (b)

that the sending computer will send the file names as well as the data. What we are concerned with is the protocol for sending the file name.

This function was added by someone other than Ward Christensen, and while it works, it doesn't look much like normal data communications practice. The basic scheme is that the sender sends the file name ONE CHARACTER AT A TIME, and the receiver ACKs each character separately. At the end, the sender sends an EOF (1AH) and the receiver replies with the checksum! If it matches, the sender sends an ACK, the receiver replies with a NAK, and we fall into the normal file transfer sequence.

<u>Sending Computer</u>	<u>Receiving Computer</u>
"Awaiting Name	"Awaiting File
NAK"	Name"
	(times out)
1st letter	NAK
	ACK
2nd letter	
	ACK
...	...
11th letter	
	ACK
EOF (1AH)	
	Checksum
ACK	
	NAK
1st Data Block	
...	...

At the end of the data transmission, the receiver sends another NAK and gets either another file name or another EOF, indicating that there aren't any more.

CRC Checking

MODEM7 also added a Cyclic Redundancy Check (CRC) option. This replaces the one-byte checksum with two bytes of CRC. CRC checking is much more reliable than checksums, giving better than 99.99% probability that any data errors will be detected.

If CRC checking is requested, the receiving system sends a 'C' in place of the initial NAK. If the sender responds

within 3 seconds, the transfer continues with CRC checking in effect. If there is no reply within 3 seconds, the receiver assumes that the sender doesn't know about CRC checking (old version or different program), sends a NAK, and settles down to use the old checksum system. In batch mode, the decision about the first file carries forward for the others.

MODEM7 and XMODEM are set up to use CRC checking by default, but the user can specify either method if he desires.

Error Aborts

The protocol will persevere through a reasonable amount of line noise, re-transmitting when necessary. If the line quality is really trash, however, after 10 retries (on the same block) the receiving computer will display "Retry or Quit?". If the operator enters "Q", a CAN (CANCEL, 18H) is sent to cancel the entire transfer session.

Problems

Unfortunately, the protocol is not entirely bullet-proof. While the following cases are very rare, they can conceivably occur:

At end of transmission, if the receiving computer misses the EOT, it will continue to wait for the next block (sending NAK's every 10 seconds, up to 10 times) and eventually "time-out".

There is a possibility that an ACK could be "garbled" to a CAN, thus aborting prematurely.

Do not use the "V" (View) sub-option during file transfer with a slow (4800 Baud or less) terminal or a hard copy printer. It takes too long to write each character to these terminals, so you don't always get back in time to catch the next one from the modem. This should be readily apparent if it occurs.

#

The Phantom Signal

By Joe Maguire

No, this is not about a message from outer space but an explanation of the S-100 bus signal found on pin 67.

To completely understand phantom, we must return to the Stone Age of micro computers (circa 1975) when just getting one to run was a major accomplishment.

History

I bet you never gave much thought as to just how the CPU gets going once you turn on the power. Well, the design of each microprocessor incorporates a "startup" mode, a condition in which the CPU signals are set to a known state. For the Z80 (and 8080 and 8085) the startup mode is initiated by first applying power or momentarily grounding the Reset pin. At that time the program counter is set to zero and all the address lines are brought low (corresponding to address 0000). Additionally, the CPU enters the "instruction fetch" state or, in other words, starts looking for something to do. It better find a meaningful instruction waiting for it at address zero or it will soon enter the twilight zone!

Now the computer designer is faced with a problem. He better have a fish ready to throw to the CPU on startup or he will never get it under control. Different manufacturers have solved this problem in different ways.

MITS (the Altair--remember?) and IMSAI used the front panel approach. On startup the front panel circuitry got a strangle hold on the CPU and held it in a wait state until the user toggled in an address or instruction by means of the myriad switches on the panel. This switch flipping got to be such a chore after a while that MITS offered a ROM-based monitor program which started at address zero and ended at address 1FFFH. Now the user could just turn on the power and hit the RUN switch. Miraculous!

But then another new garage operation just starting up, called Kentucky Fried Computers (later to become North Star),

was faced with a decision about its new disk system. Where to put the DOS? They decided on just above the ROM and so the origin of 2000H was born.

Other manufacturers such as Cromemco and Processor Technology did not use the front panel design and had to come up with another solution. Just who dreamed up the idea of Phantom I'm not sure, but I do know that Processor Tech was the first one to put it to good use in their SOL computer.

PT didn't like the idea of putting a ROM at address zero as MITS had done because the convention at the time called for operating systems to be in high memory (think "IBM" here). But how to get the CPU to jump to the ROM address which was to be at C000? The chief circuit guru at PT, Lee Felsenstein (the designer of the Osborne most recently), decided to use the phantom signal, and it worked like this:

Phantom in Action

The SOL was expected to be populated with RAM at address zero up to perhaps 3FFFH. (In those days nobody could even imagine 64K.) The ROM, at C000, could be tricked, with a little hardware magic, to appear at zero during startup but that meant it would conflict with the RAM already there. The RAM board was, therefore, designed so that it could be made to "disappear" while the ROM was superimposed on top of it. The disappearing act was accomplished with the phantom line.

Bringing the phantom line to a logic low inhibited the data output of the memory board and allowed the data in the ROM to be read instead. The ROM gained control of the CPU and everything worked fine. It worked so well, in fact, that practically every manufacturer of **memory boards** (including North Star) decided to incorporate the phantom disable line into their design.

As often happens in love, war, and computer technology, the way things were

anticipated to go, didn't. Almost no other **computer manufacturer** used the phantom signal.

North Star's Method

When North Star brought out their Horizon computer, they used a different approach.

On the North Star CPU board is an IC (location 4G) which has the sole purpose in life of jamming a jump instruction down the gullet of the CPU as it is waking up from a power-off or Reset.

When the CPU starts trying to read a byte from the RAM at address zero, the IC elbows its way to the front of the line and throws the jump instruction "fish" which the CPU reads instead of the RAM data (which is meaningless at power on anyway). The CPU then jumps to the PROM on the disk controller board and we are off to another fine start. There is no RAM/ROM conflict and phantom is not needed.

Restrictions

Knowing the history of phantom, we realize that the main purpose was to allow RAM and ROM to coexist for a few moments during startup. It can be used to allow RAM and ROM to coexist at other times but it should not be used in lieu of bank switching. Why not? Because the main criterion to prevent conflict is to have only one data source present during reading. Phantom does this by inhibiting data out of the RAM but it might not inhibit a write. It depends on the manufacturer's design. Some RAM boards just switch off the data input gates to the CPU.

You are not supposed to try to write to a ROM but Murphy's third law states that "if it can be tried, someone will try it." Knowing this, North Star's implementation of phantom completely disables their RAM boards. With some other boards, however, an accidental write can contaminate the RAM data, so trying to switch RAM boards with phantom is risky. But there is still another problem. In a bank-switching scheme one bank must

be on while the other is off. This cannot be done with phantom alone.

Can I use phantom with anything in my Horizon? Yes, and I will give you an example.

Overlaying the Disk Controller with RAM in a Horizon

Various schemes have previously been suggested in this newsletter about how to increase the available RAM, but always there is the problem of the disk controller. It sits right in the middle of the top 8K and, unless a special set of PROMs are purchased to relocate it or a RAM board with a 1K "window" is obtained, it places a limit of 56K on available RAM. But there is one other possible solution.

If the controller board can be made to generate the phantom signal, that signal can be applied to the RAM board to make it "disappear" during disk accesses. That will allow a full 64K of RAM minus the 1K used by the controller board itself. Can it be done? Yes, and quite easily too.

The CPU reads data from the controller from only two 256-byte blocks. They are the E8 and EB blocks. The remaining two blocks are addressed only to issue controller orders, not to read data. The controller already generates the signal to enable the data gates when the CPU wants to read data. All that is required is to route that signal to pin 67, the phantom line. The signal is available from pin 6 of IC 7C on the controller board. Just jumper it to pin 67 of the S-100 bus.

Egad! Pin 67 is missing!

You are a victim of economy. The later production of controller boards provided only those pins which were actually used. But there is hope. You can try to obtain "stick-on" copper pads from any source which sells circuit design aids or drafting supplies. Or, you can jumper directly from the controller board to the "PH" pad of an adjacent memory board (the pad which goes to an onboard IC, not back to pin 67). Small gauge wirewrap wire works fine.

Next, address the memory board to

include the area occupied by the diskcontroller. Now try booting the disk. You should get the normal signon message and the prompt sign. A memory test should show more available memory (up to 63K depending on what size board you are using).

Caution!

One of the hazards of using trickery is that sooner or later you are going to outsmart yourself.

When the disk controller is active, the phantom signal will disable the RAM board connected to it. With some CPU boards or

memory other than North Star's, this modification may not work.

I have tested this situation with two different types of RAM boards: North Star's 32KA (modified for 64K) and a 64K static from Digital Research (of Texas). Both worked fine.

Note that if phantom is used by other boards in your computer, or is generated by more than one source, unpredictable results may occur. I tested the modified controller board in a PT Sol and it did not work.

Give it a try; enjoy that extra RAM!

#

Dummy Lines

By Alan H. Nelson

Occasionally it is necessary, or perhaps just convenient, to print a dummy blank line in a text. I have recently encountered this need for two quite separate procedures. First, my cut-sheet feeder, which feeds single sheets into my printer, will only function properly if the print-head is centered relative to the paper at the time the paper is fed beneath the paper bail (the device with rollers which holds the paper tight against the platen). Second, MagicBind (reviewed in Compass, Vol. III, no. 4, pp. 12-14), has a back-up function for two-column printing; this function works best if a blank line can be printed at the top of the second column, so that the paper has always been travelling forward through the printer at the time of printing.

Unfortunately, many word-processing programs and "smart" printers have "motion-minimization" routines which disregard blanks or blank lines in the interest of efficiency. Take each of the two instances just mentioned.

1) Many programs will "throw away" blanks following text on a line. Thus if you put a line with thirty-three blanks in your text file in an attempt to move the head to the middle of a 66-character line (i.e. the middle of the page), the print-head may not go through this apparently

vain exercise, but, anticipating the fact that it will accomplish nothing in the way of actual printing, may stay at the left-hand edge of the text block while the printer merely executes a carriage return.

2) If you tell the printer to execute five negative carriage returns followed by five blank lines (or positive carriage returns), the program or printer may determine that this will put it back where it started without any actual printing, and the print-head may simply not move at all.

The Diablo manual proposes a solution to this problem which I have tested successfully with both WordStar and MagicBind, and may well work with other word-processing programs as well. This involves forcing the print-head to strike not merely blanks, but to strike a "dummy" character while the ribbon is lowered. Since no ink is transferred to the paper, the character is essentially invisible.

The success of this procedure is conditional on two assumptions: 1) The ribbon must be capable of being lowered (or not raised) during the impact cycle; 2) the character struck against the paper must be absolutely clean and free of ink.

Check your printer manual to determine whether your ribbon can be moved clear of the impact zone. On

Diablo printers, this procedure is related to two-color printing. With a black-red ribbon, the only two options are printing black or printing red. But if the ribbon is black-only, the ribbon will not be raised and therefore will not be struck. (Black-only printing is far more common than black-red printing!)

If you always use a carbon ribbon with a particular print wheel, the characters should be absolutely clean; thus you might choose any character as your dummy. The Diablo manual offers another solution for use with a wheel which may contain residual ink: choose as your dummy some character, e.g. the acute accent ` , which has never been used at all. (Remember, however, that most printer self-text programs run through all characters.) If you have no clean character, use a fresh daisy wheel, or clean one of your old wheels well, concentrating on the dummy character.

Here, finally, is the command line I use for WordStar with my cut-sheet feeder:

```
.he      ^Y`^Y
```

The dot command .he controls the header line (the first line on the page), in this case meant to be blank. ^Y is the ribbon-off ribbon-on toggle. ` is the dummy character. (Of course, the dummy character should be preceded by about 33 blanks, not all of which are indicated in the example.)

On print-out, the print-head dutifully moves to the center of the paper while the sheet is fed through the roller; the dummy character is struck; and the print-head remains centered while the paper is fed up under the bail. By the time the print-head returns to the left edge of the print-block for real printing, the sheet of paper is held firmly under the bail for complete control.

The Diablo manual suggests that the dummy routine should be incorporated in software routines. I was relieved to discover that it could be incorporated so easily into WordStar and MagicBind. Users with other printers, other programs, and other needs, may be able to adapt this procedure for their own purposes.

```
# # #
```

COMMENTARY

BASIC VARIABLES and QUAD DISKS

By Robert L. Porter
4005 Forge Drive
Woodbridge, VA 22193

Mr. M.Gilbert asked about a program to dump the variables in a BASIC program. An article in Dr. Dobb's Journal, no. 49 (October, 1980), provided an assembly language program to do this for single density disks (I don't know of any written in BASIC). I have altered the program to do this for double density, and can provide a copy if desired. The version I have is designed to run at 0E00H, using DOS 5.2DQ, but it can be reassembled higher in memory and run with DOS 2.1.1. I am sporadically working at modifying it further to provide a decompressed listing of BASIC and GBASIC (Advantage Graphics BASIC) program statements also, but am not sure how soon I will have that done.

When I first installed DOS 2.1.1 I had difficulty getting it to read the second side of quad disks. I have now solved the problem. It seems that if, when running SYSGEN, you answer that all the drives are quad when in fact you have only two drives, the program assumes that you have **four** drives and inserts a 0FFH into the CONFIGURATION byte. Apparently this confuses DCOM when called, and will cause errors when the DOS reaches the end of side 1 and/or tries to read side 2. The solution is to tell SYSGEN that you have "**mixed drives**", then when asked how many you can tell the program "2". SYSGEN, when told that both of these are quad drives, will place 0C3H in the CONFIGURATION byte. DOS then will read both sides of the disk properly.

```
# # #
```

Computer Pollution Solution

Add a Fan Filter to the Horizon

(c) 1984 by Steven A. Hogan

I had been using my Northstar Horizon computer for about six months when I discovered what I consider to be a general design flaw that is common not only to the Horizon, but many other computers and stand-alone disk drive cabinets as well. The problem is that the Horizon is cooled by a fan which exhausts air from the rear of the cabinet, creating a vacuum inside which draws cooling air into the box through every opening, including the disk drive slots and the unused connector cutouts in the rear panel. The operating principle is not unlike that of a vacuum cleaner, and I found that my Horizon was acting just like one. The unfiltered air being drawn into every opening of the computer was leaving behind the dust and dirt that it carried. After six months, the inside of my computer looked like an old, dusty TV set chassis, and I noticed that a film of fine dust was left behind on my diskettes if I left them for any length of time in the drives. The dirt on the S-100 cards can contribute to the problem of cooling the components on the board by preventing them from radiating heat efficiently into the surrounding air. The dirt on the disks was a disaster.

A second problem, somewhat related to the first, is the loss of cooling efficiency caused by the leakage of air through the unused connector cutouts in the rear panel. Air is supposed to enter the Horizon via a slot in the bottom and flow past the S-100 cards on the motherboard, but a great deal of air takes the path of least resistance and enters through the rear panel holes instead, thus short-circuiting the intended airflow through the card cage. As a result cooling efficiency is reduced and the fan ends up moving a lot of air that doesn't really cool the parts that need it.

This second problem was easy to fix. I obtained some black nylon hole plugs designed to fit a DB25-S panel cutout, and sealed up all the unused holes. Some 3/8"

diameter black nylon hole plugs were used to seal up the four round holes, and I made up a small aluminum plate to cover up the large unused ribbon cable slot right next to the fan under the power switch. The plate is 1" x 2-7/8" and attaches with #4-40 machine screws, nuts and lockwashers to the existing Horizon rear-panel holes. With the unused holes plugged up, air was now forced to flow past the boards, and cooling efficiency was greatly increased.

The main dirt problem was much more difficult to solve. I was at a loss as to how to cure it without installing whole-house air conditioning and an electrostatic air cleaner in my home (very expensive!) The answer came in a telephone conversation with Ron Fleming of Sanford Systems. He suggested that I reverse the direction of the fan and add a high-quality air filter to pressurize the cabinet with clean, dust-free air. This solution was so simple and yet so effective that I began to change the fan direction on my computer as soon as I hung up the phone.

Changing the fan direction was simple, but finding a suitable way of filtering the air going into the fan was not nearly so easy. A search of what was commercially available did not turn up the kind of filter that could easily fit the fan of an existing, non-filtered microcomputer or disk drive cabinet. Most of them required a drastic change in the mounting, looked ugly and were the kind with aluminum mesh that had to be cleaned and then coated with an oily filter coating in order to work right. What I wanted was a water-washable, polyurethane foam filter that could easily be maintained without tools by anyone who had access to a sink. It looked as if I were going to have to develop a custom filter.

After researching the various filter media available, I finally settled on a 1/2" thick slab of Scott foam, which is a deep-loading, water-washable polyurethane

filter medium. It is available in various porosities, and the two which seem to work best are the 30 pore-per-inch and the 45 pore-per-inch materials. The 45 pore-per-inch foam will trap finer particles, but also causes more restriction of airflow. In general I would recommend 30 PPI for most cases. The 45 PPI material should only be used with high-velocity fans, such as those found on most stand-alone disk drive cabinets. In practice, however, there seems to be very little difference in restriction between the two materials. The measured increase in temperature between the 45 and the 30 PPI foam was less than one degree C. The ambient temperature inside my Horizon rises less than two degrees (C.) with the addition of either filter, showing that neither filter causes a significant decrease in airflow.

Having settled on a filter element that made sense, the next step was to figure out a good way to mount it on the fan. The way I chose to package the filter was to incorporate the plastic finger guard that was originally supplied with the Horizon, as part of the filter. By attaching the fan and the finger guard with the screws on the inside, and by using #6-32 x 1/2" threaded brass spacers instead of nuts on the outside, the spacers provided posts to hold the filter element in place. I then had a black-anodized, brushed aluminum cover made which fit over the filter element, spacers, and finger guard to seal the sides of the filter and prevent air from leaking past the filter. The aluminum cover is held in place by four knurled thumbscrews which screw into the threaded spacers, allowing easy servicing of the filter without any tools. The exploded-view illustration shows how the whole thing goes together.

The finished filter not only looks great, but works better than I expected. Where before, every air leak in the Horizon left a deposit of dust as the air was sucked into the box, now every air leak emits a gentle breeze which prevents dust from entering. Since my filter has been in place (over a year now), I have cleaned the filter many times (filthy), and the insides of my computer are as clean as the day I first installed it. The problem with dirty

diskettes and drives has been thoroughly solved, and I couldn't be more delighted!

This fan filter is designed to work with a standard 4-1/2" tubeaxial fan. Older Horizons, like mine, used a Rotron Whisper(R) Fan Model WR2A1, which is a very quiet (but not very powerful) fan that, like all tubeaxial fans, is easily reversed. Recent production of Horizons and the Advantage do not use a tubeaxial fan (the kind that has a plastic or metal housing and built-in venturi), but have gone to an open "turntable motor" type fan that cannot easily be reversed. To use the air filter described in this article, that kind of fan will have to be upgraded (replaced) with a tubeaxial type. I upgraded the fan in my Horizon to the Rotron Whisper(R) XL model WX2H1 which delivers significantly more airflow than my old fan with only a slight increase in fan noise. (The XL series is specifically designed for low noise operation). I recommend this fan as an upgrade to those who have newer Northstar hardware.

The following parts are needed to modify any microcomputer or disk drive cabinet that uses a standard 4-1/2" tubeaxial fan:

1 ea. Rotron 550481 .26" thick plastic finger guard.

(Metal wire finger guards won't make an airtight seal to the rear panel and will not work.)

1 ea. 4.75" x 4.75" x 1/2" thick 30 PPI or 45 PPI Scott urethane foam filter element

1 ea. 4.875" x 4.875" x .812" sheet-metal filter cover with a 4.50" diameter round center hole and 4 clearance holes for #6-32 screws corresponding to the holes in the fan guard (see illustration). .062" thick 5052 Aluminum is recommended material

4 ea. #6-32 x 1/2" threaded nickel-plated brass hex spacers.

4 ea. #6-32 x 1/4" nickel-plated brass knurled thumbscrews

8 ea. #6 size internal-tooth lockwashers

4 ea. #6-32 x 7/8" pan-head machine screws (for plastic cased fans like Rotron Whisper(R) WR2A1 and 1/8" thick panels-older Horizons)

or

4 ea. #6-32 x 2-1/8" pan-head machine screws (for metal-cased fans like Rotron Whisper(R) XL WX2H1 and 1/8" thick panels-newer Horizons with replacement fan)

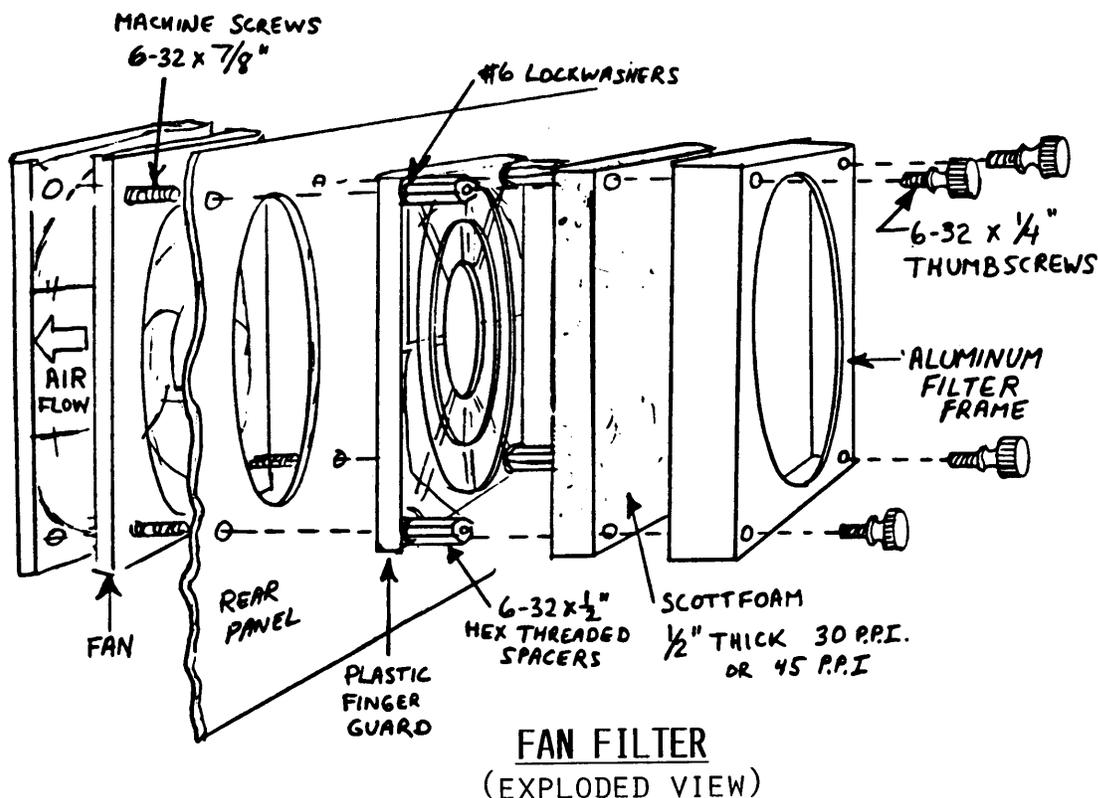
The Northstar Horizon requires the following hole plugs:

- 8 ea. black nylon DB25-S hole plugs
- 4 ea. Smith 3091 nylon 3/8" hole plugs for .125" thick panels
- 1 ea. cover plate 1" x 2-7/8" with clearance holes for #4-40 screws corresponding to holes in Horizon ribbon connector cutout
- 2 sets #4-40 hardware to attach cover plate
- 1 ea. small nylon cable ties to rebundle wires

The author will be making available complete kits containing all of the above parts to interested INSUA members and others at modest cost. Replacement die-cut filters are also available. If there is sufficient interest, the author will also make available the recommended Rotron fan for those who desire to upgrade their fans. For information contact:

Steve Hogan
The Sound Steward
1606 West Southgate Avenue
Fullerton, CA. 92633
Phone: (714) 871-6636

The following step-by-step instructions are for adding the filter conversion to the Northstar Horizon computer. Reversing the fans and adding the air filter to most other computers and disk drive cabinets would be very similar in most respects, and the average INSUA member should have no trouble adapting the procedure to his/her particular hardware. CAUTION: As with any hardware modification, this conversion may void your warranty. A word to the wise is sufficient.



Instructions for Fan Filter

1. Remove the A.C. line cord from the rear panel of the computer and unplug the terminal and any other peripheral equipment.
 2. Remove the Horizon's lid. If fan is open "turntable motor" type, purchase a reversible tubeaxial type such as the Rotron Whisper(R) XL WX2H1.
 3. Carefully cut the plastic cable tie holding the power supply wire bundle to the rear panel (located inside the computer just above the rear of the motherboard).
 4. Remove the nine #6-32 pan-head machine screws holding the back panel to the rest of the chassis. Pull the rear panel away and down, allowing access to the fan.
 5. Remove the fan and fingerguard from the rear panel. Clean the fan with a paper towel moistened with some window cleaner. Now is a good time to remove the boards and vacuum the dust and dirt from the card cage and motherboard area. Be careful not to damage anything by overzealous cleaning!
 6. Install the hole cover plate over the long ribbon cable connector hole adjacent to the fan hole beneath the power switch. The cover plate goes on the inside of the rear panel and is fastened with #4-40 hardware so that the following order results from outside to inside: #4-40 x 5/8" machine screw heads, rear panel, cover plate, #4 lockwashers, #4-40 nuts. This plate can be removed at a later time if the hole needs to be used.
 7. Turn the fan around and re-install it in the rear panel. Orient the fan so that the power connections are on the lower right corner of the computer under the power switch (Horizon only). **MAKE SURE THE AIRFLOW ARROW POINTS INWARD!** Remount the fan and fan guard so that the following order results from inside to outside (plastic cased tubeaxial): #6-32 x 7/8" machine screw heads, #6 lockwashers, fan, rear panel, fan guard, #6 lockwashers, #6-32 x 1/2" hex spacers. If you are installing an upgraded metal-cased fan, such as the Rotron Whisper(R) XL WX2H1, #6-32 x 2-1/8" machine screws must be used, and they enter from the rear of the fan instead of the center like the plastic cased fans.
 8. Carefully replace the rear panel and attach it with the nine #6-32 machine screws and #6 lockwashers previously removed.
 9. Use a new plastic cable tie to secure the power supply bundle to the rear panel. Carefully clip off the excess.
 10. Press the foam filter element over the studs formed by the hex spacers. Use an awl or an icepick to make the four holes in the corners of that correspond to the studs (see illustration). The precision diecut filter that comes with the kit already has the four corner holes -- just press it over the spacers.
 11. Place the aluminum filter cover over the filter and secure with the four #6-32 x 1/4" knurled thumbscrews.
 12. Plug all unused DB25-S rear panel cutouts with nylon hole plugs. Plug all unused 3/8" round holes with nylon hole plugs. (These plugs just pop out if the connector hole is needed later).
 13. Reconnect the A.C. line cord, terminal and peripherals.
- Periodically (at least once a month), check the filter element for an accumulation of dust and dirt. To service the filter, simply remove the four knurled thumbscrews and the aluminum filter cover. Remove the filter element and wash in warm water. Squeeze gently and blot dry with a towel. Replace the foam, cover, and thumbscrews, and you're done. The whole process takes less than 3 minutes, and should be done often. Allowing the filter to clog with dirt will obviously reduce cooling efficiency, so be sure to service it regularly, and enjoy clean computing.

#

ALIEN EQUIPMENT

By Warren Lambert

This issue's **ALIEN EQUIPMENT** was reviewed by Ted Carnevale from the Neurology Department, HSC T12Rm020, State University of New York at Stony Brook, NY 11794. He is using some sophisticated "mixed brand" S-100 boards with his North Star Horizon. If you want to learn more from Dr. Carnevale, I'd suggest reading his 1983 articles in *Microsystems*: "Put your printer on a parallel port" (July), and "Implementing the IOBYTE function on the North Star" (December).

The hardy individuals who mix boards and create a unique computer architecture often solve serious problems before they get their system running. Their gambles, effort, perseverance and intelligence benefit all of us when they develop capabilities that our "mother-knows-best" dealer never dream of.

Ironically, when Ted and I discussed his product reviews, it turned out that we had different modem programs (e.g. I don't have MODEM7), and I couldn't capture Ted's reviews; he had to copy his reviews onto my wimpy 338K 35 track CP/M, instead of his customary 388K disks, even though we both use Tandon 100-2 drives: incompatibility problems even between two Horizons!

I encountered a more bizarre compatibility problem with my junky IBM-PC clone when I took it to Computerland to find a Gee Whiz game, such as MicroSoft Decathlon. Decathlon wouldn't boot, so we tried it on every PC-clone in the store, including about five different "compatibles." Half the compatibles worked, but we couldn't get Decathlon running on an IBM-PC! About half the IBM-PC's weren't IBM-compatible, as the program would run only if you used the "right" video board on the PC. So we N* owners living in small towns aren't alone in having compatibility problems.

Enough wailing and weeping: on to Ted Carnevale's reviews of some S-100 boards he's gotten to work.

A/D and D/A interface

Product: A/D/A board
Vendor: I/O Technology
P.O.Box 2119
Canyon Country, CA 91351
Reviewer: Ted Carnevale, M.D.

This is an inexpensive but fast analog interface board with eight single-ended or four differential analog input channels, and eight analog output channels. Resolution is 12 bits and A/D conversion time is 12 microseconds. I/O can be ported or memory mapped; in the latter case, it can be PHANTOMed, or configured for 16 or 24 address lines (but watch out for N*'s grounded bus line #61, which is A20 of the IEEE 696 bus standard!--see W. Lambert's article on p. 12 of *Compass* vol. III, no. 3). D/A conversion can be to voltage or current output, and auto-refresh can be selected or disabled by software. I have been using this board for over a year now and it works like a charm. The only shortcoming of this board is that it doesn't have its own clock or TTL trigger input. Repetitive sampling therefore requires carefully written timing loops or other hardware with a clock that generates interrupts. The parallel input port on the N* Horizon can be polled or run in interrupt mode to watch for sampling triggers, in which case an external pulse generator can be used. This board comes with a BASIC program which illustrates how to use it, but it is just as easy to run with assembly language, FORTRAN, Pascal, etc. If you want fast data acquisition with an interpreted language like BASIC or a P-code interpreter like JRT Pascal, you'll have to use assembly language drivers.

(continued next page)

Multifunction Board (clock, counters, calendar, math chip, RS232 port, interrupt controllers, 4K RAM/EPROM)

Product: Systems Support Board #1
Vendor: Godbout/Compupro
Oakland Airport, CA 94614
Reviewer: Ted Carnevale, M.D.

This board has more features than I need, but it has some that I couldn't get along without. It can be ported or memory mapped, PHANTOMed or used with 16 or 24-bit addresses. I use the GKI MSM5832 calendar with battery backup to generate labels for data records (hour-minute-second) and data files (files have the extension MDD, where M = 1-9 for Jan-Sept, A-C for Oct-Dec). The Intel 8253 programmable interval timer has three timer/counters which are handy for counting TTL pulses and generating interrupt-driven timing. The 8259A interrupt controllers offer sophisticated interrupt handling, but I haven't tried this yet because my CPU card (N* ZPB-A) has its own interrupt circuitry which meets my needs. I also haven't used the RAM/EPROM option yet, but it might make a nice place to put a monitor; since it can run off the same battery as the calendar, it could also serve as a nonvolatile data buffer. The 2651 programmable UART is more flexible than the the 8251s on the Horizon's motherboard, and I have used it extensively with a 300-1200 baud modem. The major reason I got this board was because of the math processor chip (AMD 9511 or 9512 or equivalent). It accelerates floating point operations, 32-bit and 64-bit integer calculations, and transcendental functions such as sine and logarithm. Furthermore, Digital Research's Pascal MT+ comes with user-installable drivers for this chip, making it the fastest 8-bit Pascal for mathematical purposes.



IEEE 696 Static Memory

Product: Omiram 64K byte CMOS
Vendor: W/W
1771 Junction Ave.
San Jose, CA 95112
Reviewer: Ted Carnevale, M.D.

This is a very flexible static memory board. It can be bank-selected, PHANTOMed or used with 16 or 24-bit addresses. Either 8 or 16-bit data transfers can be selected. The board can be partitioned as a single 64K bank, two 32K banks, or a 48K and a 16K bank. The base address of each bank can be assigned at 0K, 16K, 32K or 48K independently. The board uses HM6116 2K x 8 static RAMs or 2716 EPROMs. It can be partially populated to "window" out selected areas which overlap with disk controllers or other memory-mapped devices. The manual is adequate, but there are so many options and jumper selections that it is a good idea to read the manual very carefully before doing anything. I have two of these boards, fully populated, and they have given me no trouble at all. I use them as auxiliary banks for temporary data buffering during on-line data acquisition. They also work with CACHE/Q, a CP/M enhancement which performs software spooling and dynamic buffering for disk operations.

Dual-sided dual density 5.25" disk drives

Product: Tandon TM-100-2
Vendor: Priority One
9161 Deering Ave.
Chatsworth, CA 91311
(and other vendors)
Reviewer: Ted Carnevale, M.D.

Although the N* disk format is hard-sectored, the mechanism and interface on the drive itself is the same as that used with many other microcomputers which have soft-sectored disk controllers, including the IBM-PC and TRS 80 (not Apple, however!). The large number of manufacturers of these drives has led to price competition which works to our

advantage. The price of Tandon and other DSDD 40 track 5.25" drives fluctuates, and every so often it drops to about \$225. They can replace the SSDD drives in the HRZ-D, doubling storage capacity and increasing speed of disk access (the head positioning mechanism moves faster, and DOS and CP/M can be configured to take advantage of this fact when you install either operating system). Installation is not too difficult if you have electronics repair or maintenance experience, but you should study the section on disk drive configuration in the Horizon Computer System Manual and Tandon's manual for instructions regarding the disk drive terminator IC and the drive address jumper. With standard implementations of CP/M for N*, DSDD drives provides 338K of storage apiece. Even so, it is difficult to fit all of the utilities I want on a single disk: Pascal MT+ with the Speed Programming Package, CACHE/Q, etc.; C80 with M80, L80, libraries and WordStar; WordStar and The Word with its enormous dictionary. Fortunately, there is a public-domain patch for NorthStar's version of CP/M which allows use of all 40 tracks (see

RCP/Ms), and I understand that S.A.I.L. Software's CP/M can also use 40 tracks (see Commentary by L. Anuta on p. 40 of Compass vol.III, no. 3). Using 40 tracks will increase usable storage to 388K per drive. That seemed like enough space for a while . . .

CALLING ALL HACKERS

If you're using **ALIEN EQUIPMENT**--hardware or software not on North Star's price list--send a review to Compass. For short reviews, send WordStar or CP/M ASCII texts to Warren Lambert, 5908 Lyons View Drive, Knoxville, TN 37919, or call O:(615) 584-1561 or H:(615) 938-3482; disks will be sent back in a week. Full length articles should be sent directly to Compass. Special kudos (the only user friendly DOS) for those who teach N* colleagues how to use RAM-disks, 16-bit CPUs, CP/M+, CPM-86 or MS-DOS, IBM-PC compatible disks, or exotic new languages (such as C, compiled Pascal, DR Logo): let fellow N* users learn from the ecstasy of victory or the agony of defeat.

#

WordStar Block Letters

By Warren Lambert

The headline for the "Alien Equipment" column in this issue was printed with ordinary WordStar and a Nec Spinwriter, a daisy wheel printer able to move its carriage in small increments. Qume and Diablo make similar printers. The letters were 9x7 block letters which I made with a JRT Pascal utility program; I've also made them with SAS Proc Explode, capturing them by modem and then "composing" words with column-block moves in WordStar (my WS3.0 dies if the cursor goes off the line, so block moves are tricky.) You can also type the letters out manually on the screen, a tedious procedure.

While the WordStar manual says that you can't change line height (.LH) within a manuscript, I found that Mailmerge is

smart enough to do the job if you're really careful in setting the title up.

The word "Alien" is printed below directly from the column. The column appears:

```
.lh8
.cw10
.po31
.he
.fo
.mb6
.mt3
.fi alien=.blk
.lh8
.cw10
.po31
.he
.fo
.mb6
.mt3
```



The inserted file is as follows (ALIEN is trimmed to fit into a single column):

.cw12
.po26
.he
.fo
.mb4
.mt2
.lh 2
.cw 3

```

=      ==      =====
=====  ==      ==      ==
== ==   ==      ==      ==
==  ==  ==      ==      ==
=====  ==      ==      =====
==  ==  ==      ==      ==
==  ==  ==      ==      ==
==  ==  ==      ==      ==
==  ==  ==      ==      ==
==  ==  =====  =====  =====

```

.lh8
.cw12
.po26
.he
.fo
.mb4
.mt2

The redundancy of the dot commands helped this fragile procedure work. Once you get a file right, the technique is robust and dependable, as I make "letterhead" this way for boilerplate letters.

Centering titles is complicated by the fact that the page offset (.PO) refers to the most recent character width, or the default if none is given; forgetting this fact causes grief when you change character width, and may make you think WordStar is crazy.

I put the extra LH-CW specifications around "ALIEN" in the external file in because WordStar's page length counter is fragile when you change line heights in the middle of the stream, and this undocumented feature seemed to work better when I went back to normal width while inside the merged external file. Without Mergeprint variable line height won't work at all.

The biggest thing to remember when fooling around with undocumented or verboten WordStar features is that this mature program is not crazy, and if it gives bizarre results, it may be doing exactly what you asked (if not what you wanted).

#

PORTABLE ADVANTAGE! PORTABLE ADVANTAGE! PORTABLE ADVANTAGE!

(The following notice was originally published in the January 1984 Polaris, the newsletter of the North Star Computer Society, P.O. Box 311, Seattle, WA 98111. The note was supplied by the Puget Sound Computer Service. --Ed.)

The big news for North Star users emanates from Seattle. After three months of labor by Drev, PSCS proudly announced the birth of the "PORTABLE ADVANTAGE by PSCS." Herb took it to COMDEX to show it to North Star folks, distributors, and dealers. Everyone, including Chuck Grant, North Star president, reacted favorably. We are now accepting orders, and will start production early in the second quarter.

For those who did not see it at the EXPO, the Portable Advantage consists of a North Star Advantage motherboard and keyboard combined with a 9 inch monitor and two half-height disk drives, all packaged in an aluminum case. It will be available with all normal Advantage options plus an internal 10 meg hard disk.

For information, write:

Puget Sound Computer Service
3940 Wallingford North
Seattle, WA 98103 (206) 632-6211

Beyond 64K

By Joe Maguire

The Z80 CPU and its kinfolk of other eight bit processors have 16 address lines which can define 65536 combinations of high and low logic signals. This translates to a directly addressable memory range between zero and FFFF.

If we wish to extend the capability beyond the 64K of direct addressing, we have to resort to various circuit tricks. Often mentioned are terms such as bank-switching, extended addressing, and memory management. Just what these terms mean will be the subject of this article.

Bank-Switching

A memory "bank" can be thought of as one contiguous chunk of RAM. It could be as small as one byte or as large as several megabytes. With eight bit micros it is generally considered to be the capacity of one or more memory boards. The important distinction is: a bank is one unit; something to turn on or turn off or manipulate in some way.

The theory behind bank-switching is somewhat analogous to putting a new sheet of paper in a typewriter. As one page is used up we insert another.

In a computer environment, we insert a new page by switching to another memory bank. The new bank has the same memory address as the old one. We avoid conflict by having only one bank turned on at a time. This is accomplished by a combination of hardware and software.

The most common method of switching is by means of an I/O port. The proper value is loaded into the A register and then an OUT instruction is executed. The switching circuitry on the memory board(s) decodes the data byte and sets or resets an IC latch (flip-flop). The latch is tied to the board enable/disable line. This is the method used by North Star. North Star uses port C0 and data bit 0 for bank-switching. Unfortunately, North Star's manuals leave out much important information on how to implement such a

scheme. A separate article in this series will deal with that problem.

Extended Addressing

If bank-switching can be thought of as changing sheets of paper in a typewriter, then extended addressing is analogous to slewing to top-of-form on your printer. In a real sense the new page or bank is still joined to the old one.

Extended addressing is a result of technological growth. As the S-100 bus was expanded to accommodate 16 bit processors, it became obvious that more address lines were needed.

CPUs such as the 8088 can directly address up to one megabyte of memory (actually 1,048,576 bytes) using 20 address lines. The 68000 CPU can address almost 17 megabytes by means of 24 address lines. The newly defined IEEE 696 standard for the S-100 bus provides for these additional lines by assigning new pin numbers for them. Extended addressing is the term used to indicate the use of these additional lines.

Most newer designs of memory boards have provision for extended addressing through one means or another.

Using Extended Addressing

If a memory board capable of extended addressing is used with a CPU having only 16 address lines, the CPU can be tricked into believing it has more than 16 lines by a process similar to bank-switching.

A latching IC, with its outputs connected to the extended address pins of the S-100 bus, is sent an address byte by means of an OUT instruction. Then the CPU addresses a memory location using its normal 16 address lines. The combination of the latched bits on the extended lines plus the 16 bits provided by the CPU fool the memory boards into thinking that an address above 64K is wanted.

Notice, though, that the success of this trickery depends on all of the memory

boards in the system being able to recognize an address of more than 16 bits. This is necessary because the memory in the lower 64K must be able to tell when the upper 64K is being addressed. It does this by monitoring one or more of the extended address lines.

Also, some additional circuitry must be present somewhere to do the required latching. This brings us to the memory manager.

Memory Management

Few of the older eight bit CPU or memory boards contain the circuitry required for extended addressing. To meet this shortcoming, the memory manager boards were introduced. They can range from the simple to the ultra sophisticated.

A simple manager might consist of only a few ICs and is sometimes included right on the CPU board. It would perform only the minimum function of bit latching.

The most powerful type is housed on its own board and may provide such things as: multiple switching ports, multiple latches, dedicated RAM, ROM based switching routines, and sometimes even its own CPU to handle all the overhead.

Is all this necessary? Maybe. It depends on the particular application.

With the introduction of systems such as the IBM PC and North Star Advantage, with their more powerful processors, higher performance software is becoming available. The natural inclination is to try to keep abreast of the trend. However, chucking the large investment most prior users have in their older equipment just doesn't make sense. Hence, some manufacturers, such as CompuPro, have made available boards with multiple CPUs which allow the use of familiar software as well as software designed for the newer generation of processors.

The sophistication of the coprocessor, the amount of older generation boards in the computer, and the type of software to be used, all determine what the requirements of the memory manager should be.

When to Use What

Bank-switching is a scheme which can be used if the system is composed entirely

of older generation CPU and/or memory boards. In most cases the necessary hardware is already there. Only the software need be modified.

Extended addressing is the thing to use if the system is composed entirely of a mix of old and new technology. A CPU such as the 8088 can use extended addressing without any circuit tricks but, if it is sharing the computer with another processor such as a Z80 or 8085, that CPU will need the help of a memory manager in order to make full use of the memory.

What about using the phantom signal for bank-switching? It won't work. Phantom is intended to be used to switch memory off when ROM will be overlaid upon it. **A separate article in this series, "The Phantom Signal," will explain the reason for the restrictions but also will give an example of how phantom can be used to good advantage.**

Requirements

Thus far we have only discussed definitions. The actual implementation of a bank-switching scheme must conform to some strict rules if it is to work successfully. I have already touched on rule #1.

Rule #1: Only one memory bank at a time can occupy the same address.

The reason for this should be obvious. More than one set of data available at the same address would just cause confusion.

A bank-switched environment places certain requirements on any software to be used with it. Many routines which will work fine when used with direct addressing will not work at all when memory switching is going on.

Consider this: An application program is running in a bank-switched environment. A routine in bank "A" wants to call a subroutine located in bank "B". The calling routine must first store the return address on the "stack" (a reserved memory location). Next, it jumps to the subroutine in bank "B" (after switching) and starts to work. But what about any data it needs in bank "A"? And what happens when it's time to return to the main routine? The normal return process "pops" the return address off the stack.

But the stack is back there in the other memory bank! I think you can see that things can get out of hand quickly. This brings us to rule #2.

Rule #2: All references from one bank to another must be funneled through the switching routine.

The switching routine must make provision for all the loose ends and can quickly become very involved. This is why multi-user software is expensive. It frequently uses shared memory. The switching routine is essentially a juggling act. It must maintain control no matter what. Now we're ready for rule #3.

Rule #3: The switching routine must not switch itself.

That would be like pulling the rug out from under itself. Now you see that in addition to software restrictions, we have some hardware restrictions as well. Some piece of RAM somewhere in the system must never be switched. That RAM could be on the memory manager board.

By now you must be thinking that the situation is hopeless. Not so! It just so happens that some of the software you are no doubt using every day "accidentally" conforms to the rules. This includes the DOS/BASIC combination and CP/M.

CLIP TIPS

Joe Maguire

The Three Laws of Bank Switching

Law #1: Only one memory bank may occupy the same address at the same time.

Law #2: All references between banks (jumps, calls, data, etc.) must pass through the switching routine.

Law #3: The switching routine may not switch itself.

Software

Have you ever noticed that when you are using North Star BASIC, you must also have the DOS loaded? Of course! Whenever BASIC wants to do some I/O or disk access, it "funnels" its requests through the DOS. This makes the combination a natural for bank-switching. The same goes for CP/M. A properly written CP/M program "funnels" all requests through the BDOS entry point at address 0005H. I say "properly written" because some CP/M programs use programming tricks to call the BDOS or BIOS directly and these will not work in a bank-switched environment.

Conclusion

Bank-switching or extended addressing can be a useful technique to increase the power of your system. If the restrictions are understood and the rules followed, no difficulty should be encountered.

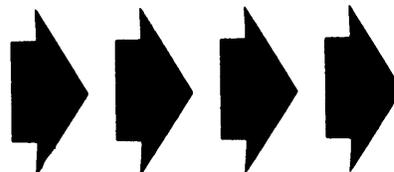
A further article in this series will explain how to implement a bank-switching scheme on a Horizon computer.

###

MEMTODAT

By Leonard Morgenstern

MEMTODAT is an improved version of an existing program which will construct a file consisting of DATA statements representing the ASCII values of bytes in memory. The first DATA statement will consist of the first and last addresses, and the succeeding lines contain ten items per statement. (Two lines near the beginning of the program need to be personalized prior to use. They specify the clear-screen character for your terminal and the address of the BASIC actually being used.)



```

10 REM MEMTODAT
20 REM CONVERT RAM CONTENTS TO DATA STATEMENTS
30 REM BY LEONARD MORGENSTERN/ UPDATED 9/25/83
40 REM == PERSONALIZATION
50 CS=CHR$(11) \REM CLEAR-SCREEN CHARACTER
60 B9=256 \REM ORIGIN OF BASIC
70 REM ==
80 F$="????" \L$=1000 \M1$="0" \M0$="0" \L1$="10"
90 DIM SS(100),I1$(20)
100 !CS, THIS PROGRAM WILL CONVERT BYTES IN RAM TO BASIC"
110 !"DATA STATEMENTS."
120 !"DO YOU NEED INSTRUCTIONS ", \ ON FNY(0) GOTO 130,140
130 GOSUB 1130
140 !CS, \For each parameter, press CR if ok, or insert ",
150 !"desired value"!
160 !TAB(30), "CURRENT", TAB(50), "CHANGE TO"
170 !"File name,unit", TAB(30), F$, TAB(50), \INPUT1", I1$
180 IF I1$<>" THEN 190 \I1$=F$ \I1$
190 IF F$=I1$ THEN 210
200 F$=I1$
210 IF F0<>0 THEN 290
220 F1=FILE(F$) \IF F1>0 THEN 290
230 !\ \TAB(5), F$, " does not exist. Will be created"!
240 !"Limits of data to be stored in hex "
250 INPUT "Lower: ", M0$ \INPUT " Upper: ", M1$
260 I1=INT((FNH1(M1$)-FNH1(M0$))/256)+2
270 CREATE F$, I1
280 GOTO 140
290 F0=1 \IF FNP(0)<>0 THEN !
300 !"Start data line number", TAB(30), L1$, TAB(50),
310 INPUT " ", I1$ \IF I1$<>" THEN 320 \L1$, \I1$=L1$
320 ERRSET 330, E1, E2 \L=VAL(I1$) \GOTO 340
330 ! " ??? " \GOTO 300
340 ERRSET \L$=I1$ \!
350 !"Data line interval", TAB(30), L1$, TAB(50),
360 INPUT " ", I1$ \IF I1$<>" THEN 370 \L1$, \I1$=L1$
370 ERRSET 380, E1, E2 \L1=VAL(I1$) \GOTO 390
380 ! " ??? " \GOTO 350
390 ERRSET \L1$=I1$ \!
400 !"Starting memory address (hex)", TAB(30), M0$, TAB(50),
410 INPUT " ", I1$ \IF I1$<>" THEN 420 \M0$, \I1$=M0$
420 M0=FNH1(I1$) \IF M0<0 THEN 430 ELSE GOTO 440
430 ! " ??? " \GOTO 400
440 M0$=I1$ \!
450 !"Ending memory address (hex)", TAB(30), M1$, TAB(50),
460 INPUT " ", I1$ \IF I1$<>" THEN 470 \M1$, \I1$=M1$
470 M1=FNH1(I1$) \IF M1<0 THEN 480 ELSE 490
480 ! " ??? " \GOTO 450
490 M1$=I1$ \!
500 !"OK? ", \ ON FNY(0) GOTO 510,160
510 \OPEN#0&F1, F$
520 SS$=" +CHR$(143)+ " MEMORY DATA STATEMENTS "+FNH$(M0)
530 SS$=SS$+ " TO "+FNH$(M1) \GOSUB 850
540 SS$=" +CHR$(135)+STR$(M0)+ " , "+STR$(M1) \GOSUB 850
550 FOR M=M0 TO M1 STEP 10
560 SS$=" +CHR$(135)+ "
570 IF M+9>M1 THEN N1=M1 ELSE N1=M+9
580 FOR N=M TO N1
590 SS$=SS$+STR$(EXAM(N))
600 IF N<>N1 THEN SS$=SS$+ " , "
610 NEXT
620 GOSUB 850
630 NEXT
640 WRITE#0, &I
650 !
660 !"The file ", F$, " should be set to type 2 and its length"
670 !"byte set by running the program RESTYP2"
680 !"END OF JOB" \ END
690 REM convert hex to decimal, return -1 on error
700 DEFFNH1(I1$)
710 I1=LEN(I1$)
720 IF I1=0 THEN RETURN -1
730 IO=0
740 FOR I2=1 TO I1
750 I3=ASC(I1$(I2,I2))
760 IF I3<48 OR I3>70 THEN EXIT 810
770 IF I3>57 AND I3<65 THEN EXIT 810
780 IO=IO*16+I3-48-(I3>=65)*7
790 NEXT
800 RETURN IO
810 RETURN -1
820 FNEND
830 REM WRITE A DATA LINE
840 REM ( LINE#=L, STRING=SS )
850 I1=INT(L/256)
860 WRITE#0, &(LEN(SS)+4), &(L-I1*256), &I1
870 FOR I=1 TO LEN(SS) \ WRITE#0, &ASC(SS$(I,I)) \ NEXT
880 WRITE#0, &I3
890 !L, \L=L+L1
900 RETURN
910 REM FUNCTION TO INPUT Y/N . RETURN 1 IF Y, 2 IF NOT Y
920 DEFFNY(I)
930 INPUT "(Y/N)? ", I1$
940 IF I1$=" " THEN RETURN 2
950 I1$=I1$(1,1) \ IF I1$="Y" THEN RETURN 1 ELSE RETURN 2
960 FNEND
970 REM DECIMAL TO HEX CONVERSION
980 DEFFNH$(X)
990 I1$="" \ I3=X
1000 FOR I1=LTO 4
1010 I2=INT(I3/16) \ I1$=FNH1$(I3-I2*16)+I1$
1020 I3=I2
1030 NEXT
1040 RETURN I1$
1050 FNEND
1060 REM CONVERT VALUE TO SINGLE HEX CHAR
1070 DEFFNH$(Y)=CHR$(48+Y+(Y>9)*7)
1080 REM NEXT FUNCTION RETURNS PRINT POSITION
1090 DEFFNP(U)
1100 I1=B9+17 \ I1=EXAM(I1)+256*EXAM(I1+1)+U
1110 RETURN EXAM(I1)
1120 FNEND
1130! This program will construct a file"
1140! consisting of DATA statements representing the ASCII"
1150! values of bytes in memory. The first DATA statement"
1160! will consist of the first and last addresses, and the"
1170! succeeding lines contain ten items per statement."
1180! Two lines near the beginning of the program need to be"
1190! personalized prior to use. They specify the"
1200! clear-screen character for your terminal and the"
1210! address of the BASIC actually being used."
1220 !
1230 !"PRESS ANY KEY TO PROCEED" \ I1$=INCHAR$(0) \ RETURN

```

Vendors' Column

In this column we reprint vendors' descriptions of their own products. Of course, INSUA makes no claims about the quality or merchantability of any commercial software or hardware. **Caveat emptor!**

The Printer Works' new Intelligent Printer Interface/Buffer, IPIB for short, is a single-board Z-80 microcomputer designed to be a versatile enhancement for the Diablo 1610, 1620, 1640, 1650, and 630 model printers. The IPIB-IPIB-1600/630 lets your printer receive up to 64K bytes in just seconds, from any serial or parallel source, and frees your computer for other functions while print-out is completed.

And the IPIB slots right inside your Diablo printer--no external box, or power supply is needed.

In addition to high-speed data buffering the IPIB-1600/630 offers all of the following features:

- * Standard RS-232 serial interface, with switch selectable baud rates up to 19,200 baud, and full hardware and software handshaking.

- * Parallel interface for direct connection to either an IBM PC parallel printer adapter or a Centronics parallel type computer output port.

- * Buffer clear, and print pause using the printer's control panel.

- * Installs easily and in just minutes right inside your printer.

- * Surprisingly low cost. (Current cost is \$295.00.)

More Facts about The New IPIB 1600/630:

- * The IPIB can be connected to two computers simultaneously.

- * The IPIB allows IBM PC user's to send data to the printer via the parallel printer adapter. This is important because almost all of the software programs for the IBM PC expect the printer to be attached to

this parallel port, not the asynchronous communications adapter.

- * All features of your Diablo, such as print enhancements controlled by your word-processing program, are retained.

- * Internal self-test performs memory test on the RAM buffer and sends a line containing all printable ASCII characters to the printer.

- * Baud rate, parity type, stop bit count, and self-test are all switch selectable.

- * Limited one year factory warranty on parts and labor.

- * Custom programming available for special applications.

- * Dealer inquiries invited.

To find out more about the new IPIB-1600/630 just put your name, address, and telephone number, on a postcard and mail to The Printer Works, 1961 Alpine, Hayward, CA 94545, or call The Printer Works at (415) 887-6116.

\$ \$ \$

N*CAT and ADBAUD

Having been a dedicated Northstar user since Horizon S/N1997, I have longed for something like SWEEP to permit selective batch file copying on NorthStar DOS disks. Finally I gave up looking and wrote N*CAT. The program uses DOS calls to do the work and can handle quad catalogs with 128 files in 2 screen pages. --Charles Makinson

N*CAT is a disk catalog editor that eliminates the need to successively run the DOS commands RE, DE, and CF for

multiple files. The files to be renamed, deleted, or copied are selected from a four column catalog display and processed as a batch. Works with double or quad drives and DOS 5.2, 2.1 or GDOS 2.0. Price: \$29.

ADBAUD will change the baud rate on the **ADVANTAGE**. Just type:

ADBAUD xxxx

where xxxx is the desired rate. An excellent companion to **MODEM7**. Use **CPMGEN** only when the rate needs to be permanent. For CP/M systems. Price: \$15.

Write: TSR Microcomputer Consultants
PO Box 9191
Akron, Ohio 44305

\$ \$ \$

AUTOTYPE for your Horizon

AUTOTYPE is a full-featured editor/text processing system for North Star 5.x and 2.1x DOS, Z-80, 32K and double-density. **AUTOTYPE** runs on the North Star **HORIZON**. Included with the text processor are several utility programs--file rename, alphabetize directory, etc. Several North Star utility programs such as file copy, disk copy, and disk compact are also used. These utilities are normally menu selected from within the word processing system, but all are also runnable independent programs which can be used separately for other purposes.

- * 225 page instruction manual.
(Includes hundreds of examples)
- * 120 full screens of on-line HELP.
(15,000 words of instant HELP)
- * Automatic startup on RESET.
- * Relocatable to any memory address.
- * Easily configured for any terminal.
- * Automatic drive search for filenames.
- * Useful and friendly error messages.
- * Compatible with PDS assembler, Small-C
- * Also edits North Star BASIC programs.
- * Complete search and exchange. TABs.
- * Block move, Block copy; Merge files.
- * Print unlimited length document.
- * Repeat printing of a file.
- * Dynamic control of all margins.

- * 7 page number styles, 8 locations.
- * Form letters, inserts during printing.
- * Dynamic control of your printer.
- * Teaching program for new users.
- * 29 programs, utilities and files.
- * Distributed on two disks.
- * Quick reference card.

AUTOTYPE is usable by a beginner in a few minutes using only a few simple commands; but provisions have also been made for creating, editing and printing complex computer programs and extensive documents of all kinds.

The distribution disks include one copy of an immediately runnable version, plus an interactive **CONFIGURATION** program and relocatable files to create a custom version that exactly matches your computer system. The system can be relocated in memory to match your particular version of DOS, and can be customized to match your CRT terminal. Friendly and easy to use: just insert the **AUTOTYPE** disk and **RESET** for an automatic startup.

- * Master control menu schedules the programs needed for your job.
- * Automatic drive searching finds your file on any disk drive.
- * Filenames are automatically passed from one program to another.
- * All features fully described in on-line HELP system within EDITOR.
- * Disk directory always available to help find the right filenames.
- * Useful and friendly error messages describe any problem.

HARDWARE REQUIRED: North Star **HORIZON** computer (or compatible NS **MICRO-DISK** system, Z-80 only) 32K minimum RAM memory. More is recommended. At least one double density disk drive. Two or more drives are recommended.

SOFTWARE REQUIRED: North Star DOS, release 5.1, 5.2, or 2.1.x; North Star **BASIC** (required only for **TEACHER** program); North Star Utility programs: **CD**, **CF**, **CO**

Bargain price! \$39.95 + Shipping.

Write: DSE, PO Box 40174
Albuquerque, NM 87196
(505) 266-2491 after 6 pm MST

Appleby on TurboDOS

By Don E. Appleby

I received the TurboDOS operating system as part of the "Flexi-bundle" software packaged with the new Horizon which I purchased in December of 1983 from my local dealer, Bob Tourula of Microcomputer Solutions. I also received Wordstar and DBaseII in the "Flexi-bundle" package.

I opted for the 15 MB hard disk. This package retailed for \$6,699.00, although I received some concession as a result of my many years of patronage of the store owner's various North Star Computer businesses.

I selected the multi-user Horizon and TurboDOS because of their compatibility with my present systems. I also have another Horizon (serial number 00045), an Advantage 8/16 and an Osborne I (tan case).

I operate the Bankers & Hackers Bulletin Board System and RCP/M on the older Horizon. But I was getting jealous of all those callers being able to use my computer to the exclusion of me. I wanted to be able to use it at the same time.

I had considered upgrading it with TurboDOS and a hard disk, but the cost of a new computer was very little more. So I dove in head first, much to the pleasure of my dealer, banker, and, no doubt, Mr. Charles Grant as well.

It is well I kept the old faithful SN#00045. "Alas" as Jerry Pournelle says, the new Horizon and TurboDOS are not set up for communications (at least not as configured in the North Star 2-user set-up). As yet I have not concerned myself over it, since I have been too busy learning about TurboDOS and some of its incompatibilities with CP/M software.

I am still trying to get hold of a technical manual from North Star for the HRZ-UP8 user boards that came with the new Horizon. It would also be nice to have the source code for the user.asm for TurboDOS (or whatever is comparable for TurboDOS as opposed to CP/M).

I use and love the ZCPR console

command processor replacement (though not on the TurboDOS system) so generously donated to the public domain by Messrs. Richard Conn, et al. Other public domain software including the Library Utility and related programs are invaluable to me. As more people use and become acquainted with TurboDOS, these public domain programs will take into consideration the useful vagaries of TurboDOS, chief of which is the increased number of user areas available (16 through 31). Many of my favorite programs won't work in those user areas if they require file handling, since CP/M, for which they were written, does not know about user areas 16 through 31.

Another bothersome attribute of TurboDOS is that it is a hog of the hard disk. It won't tolerate any other operating systems having files on the Hard Disk with it, unlike the very lenient HDOS, which on my Advantage permits CP/M and MS-DOS files plus the HDOS files and accounts as well. It also permits the 15 megabyte hard disk to be allocated into no more than two logical drives, A: and B:. This makes for some rather awkward directories when a lot of files are listed. TurboDOS, unlike CP/M, does not have "SYSTEM" files. Instead, it uses the "SYS" attribute in the file directory for the Global attribute. This permits a file in user area 1 to be used from another user area by "Privileged" users, similar to ZCPR's facility of searching A0: for commands not found on the currently logged drive/user area. There are some differences, though, in that some programs using overlays may also search for the overlays in user A0:.

Until I recently purchased Borland's Turbo PASCAL, I had no applications programs for MS-DOS to run on my Advantage. And since you can't really consider Turbo PASCAL's compiler an application program in the sense of WordStar, SuperCalc or DBaseII, I still don't have any application programs for MS-DOS, other than the DEMOCALC that

Turbo PASCAL provides as a demonstration program for its compiler. It is a "simple" SuperCalc-type program. But since I have SuperCalc-80 I will use the CP/M-80 program, rather than the DEMOCALC in MS-DOS.

The paucity of programs for MS-DOS on the Advantage seems to be a result of 1) the more than adequate use I derive from CP/M-80 programs and 2) the incompatibility of programs such as Lotus 1-2-3 and SuperCalc-3 with the North Star operating systems. Lotus and other IBM-PC programs unfortunately make calls to the IBM proprietary ROM routines, rather than keeping to the generic MS-

DOS functions. I can't justify spending the money on duplicate programs just to use my MS-DOS. And so far, my data files fit comfortably in the 64K memory address capability of the Z80 CPU.

What would be useful for the Advantage would be a software package that would utilize some or all of that 128K of memory I purchased for the 8088 CPU, as an electronic drive, ala "Semi-Disk". Unfortunately, since North Star doesn't have the 500,000+ user base of IBM or Apple, no ambitious software firms I know of have written such a program for the Advantage.

#

Advantage Graphics

The following non-North Star vendors have software which runs under Graphics MS-DOS on a North Star ADVANTAGE. (Information from February, 1984 Polaris, newsletter of the North Star Computer Society, P.O. Box 311, Seattle, WA 98111.)

Computer Innovations
75 Pine Street
Lincroft, NJ 07738
"C" compiler

CYMA Corporation
2160 East Brown Road
Mesa, AZ 85203
General Accounting,
Medical, Construction
Client Accounting

Disco-Tech
P.O. Box 1659
Santa Rosa, CA 95402
Civil and Structural
Engineering

Lexisoft
712 5th Street, Suite B
Davis, CA 95616

Select Information Systems
919 Sir Francis Drake Blvd.
Kentfield, CA 94904
Word Processing

Solid Software
5500 Interstate N. Pkwy., Suite 501
Atlanta, GA 30328
General Accounting
Vertical Markets

Sorcim
2310 Lundy Ave.
San Jose, CA 95131
Spreadsheet, Word Processing,
Spelling Checker

Supersoft Associates
1713 Niel
Champaign, IL 61820
"C" and FORTRAN compilers,
Word Processing

Micro-AP
7033 Village Parkway, Suite 206
Dublin, CA 94566
Data Base management

Microsoft
10700 Northrup Way
Bellevue, WA 98004
Spreadsheet

Star Computer Systems
18051 Crenshaw Blvd., Suite D
Torrance, CA 90594
Legal

T/Maker Company
1742 Willow Road, Suite 206
Palo Alto, CA 94304
Integrated Word Processing,
Spreadsheet and File Management

Ribbonizer Revisited

By Alan H. Nelson

In Compass III, no. 1, pp. 29-30, I gave a favorable review to **Le Ribbonizer**, manufactured by Ben Torres Ribbon Service, Box 1727, Redlands, CA 92373. Since that time I have experienced one major and one very minor problem with this re-inking device. I drafted the following note, and sent a copy to BTRS:

Although I am still impressed by this product and use it frequently, I have had two problems of which the potential buyer/user should be aware.

The first is that the ink comes in a bottle with a plastic cap which is subject to deterioration and thus leakage. Due to my lack of caution, two different bottles which I left stored on their sides ended up leaking ink with near disastrous consequences. Fortunately, I have only the blackened bottom of a desk drawer to remind me of the leakage; with a little less luck, I would have had a badly stained office rug and/or wooden floor. I now keep the ink bottles sealed tight in a glass jar with a lid.

The second problem is that the motor which drives the re-inker tends to heat up to the point where it could cause a painful burn. I have not actually suffered any untoward consequences in this respect, but I now keep a close eye on the machine just in case it should seriously overheat.

I still think this product is a good one. The manufacturers should consider providing a more durable cap for the ink bottle, and perhaps a motor which would be less subject to strain and overheating.

Within a week of mailing the note, I received a prompt reply from Mr. Ben Torres, and a few days later, an even longer letter from the manufacturer of **Le Ribbonizer**, Terry L. Westbrook, A-West, 401 North La Cadena Drive, Colton, CA 92324. Since Mr. Westbrook's letter is a model of responsible end-user support, and contains valuable information besides, I reproduce it in full:

Dear Mr. Nelson:

Thank you very much for your feedback about **Le Ribbonizer**. I am glad to hear that you use it frequently and that your opinion is generally favorable.

I am sorry that you were unlucky enough to receive two bottles of ink which were fitted with caps which deteriorated. Two replacement caps are enclosed. A bottle supplier inadvertently shipped us caps made of the wrong material.

I would normally end my discussion here, but I thought you might be interested in this additional information: So far as I know, we have not delivered any ink with the wrong caps for about a year. As you can see, the "good" caps and the "bad" caps look alike. So there is no way for a person to tell which they have, unless it deteriorates. For a while we tried some white and yellow caps from a different manufacturer, with a slightly different shape. They are of a proper material, but did not always seal properly.

We go to great trouble and expense to provide the customer with this particular bottle, controlled dropper tip, and overcap. You would expect to have "no problem" from a simple thing like obtaining bottles, right? Not so. First, the items are not stocked by the manufacturer--they are made to order in quantities of no less than 25,000 pieces ... payment in advance and delivery up to six months later.

So when we need a bottle part we literally scour the country looking for odd lots of the particular item we need--regardless of price. The problem is particularly bad with the four-ounce bottle, because the small neck is not normally found on such a large bottle.

Because of these problems, it may someday be necessary for us to package the ink in a four-ounce bottle with a "yorker spout." That is a long, pointed cap with a little red cover that just pushes on. To keep it from leaking during shipment, the spout would have to be "blind" (no hole). The customer would have to puncture the tip with a pin, which

could be messy if not done expertly. Then it could be used directly, or used to refill the small bottle having the controlled dripper tip--another messy operation. Or, the customer could buy another small bottle for \$2 instead of \$6 for four times as much ink.

Perhaps you can see why we don't particularly like the alternatives, from a customer's point of view. Therefore, we will continue to supply the ink bottle with this unusually high quality cap, as long as we can get them.

About the motor heat: The motor is rated at three watts. So, it has about three watts of heat to dissipate. That is about the same as a Christmas tree light bulb, which is usually four watts. Because the motor is larger than the light bulb, it will not get as hot as the Christmas tree light bulb.

In testing, we have stalled the motor and left it on for hours. It did not burn out. We have never seen a circumstance where the heat was enough to soften or deform the plastic ribbonizer base.

Of course there are high-efficiency motors that could produce the needed power while producing less heat. They are usually stepper motors or low-voltage DC motors. They usually do not have a built-in gearbox. They would need a power transformer, and rectifier or control circuitry or batteries. My guess is that if I built the ribbonizer using such technology instead, the price would be around \$100. To me it seems too high a price to pay for reducing wasted heat.

With this type motor, the amount of heat does not have much to do with the amount of strain. If the strain is too much, it simply stalls. We have built a few custom ribbonizers, requiring more power to advance the ribbon. We have a special gearbox for an extra \$10 which provides the extra power, but it runs a lot slower. So it takes longer to ink the ribbon. For our regular line of inkers, we think the tradeoff between power and speed is optimum.

By the way, we also have motors with 220/240 volts. They are an extra \$10. Most of our foreign customers (except Canada) need this motor. These orders should be placed directly to A-West, and

not to our exclusive domestic distributor, BTRS.

Again, thank you for writing and for this opportunity to furnish additional information.

Sincerely,
Terry L. Westbrook

* * *

The same mail that brought Mr. Westbrook's letter also brought the following note from Saul Levy:

* * *

I can add a little to the discussion of printer ribbons. I bought six reload loops for my NEC 8023 from Ben Torres Ribbon Service. I received the loops by return mail. Loading them is a little messy, but not difficult (the instructions included are adequate). I had a little problem getting the Mobius twist to stay in the left arm of the cartridge. The biggest problem was getting my rusty can of hand-cleaner open! The first loop I used lasted for five months of light to moderate usage. The fabric showed little wear. The original NEC ribbon was used for six months and showed no wear at all. These could have been reinked, but I don't have a reinker and don't want to keep a messy ribbon around until I do have one!

#

Quic-n-easi

By Leland C. Swenson

A product I have recently purchased may be of interest to Advantage users less skilled in computer programming (like me). This is the "Quic-n-easi Applications Generator/Report Generator" software which is priced at under \$300.00 in North Star version and which is a hybrid between canned programs and high level languages. It makes writing programs relatively easy for non-programmers (but not as easy as the ads say). After two weeks of frustration I was able to design a very satisfactory client and expense record and basic accounting package which runs well and can be modified for many uses.

User's Visit to North Star

By Saul G. Levy

On April 29, 1983 I visited the North Star plant in San Leandro, CA. I have several comments to make about North Star's user support.

The Compass has published several comments about this topic (see Vol. II, No. 3, pp.8-9 and 33-4). I feel that user support is North Star's biggest problem. Its hardware and software are mostly excellent to outstanding, but just try to get some support! I have been unhappy about this for a long time even though I don't need support from them very often. My latest visit confirmed that North Star won't supply any service at its plant. This is a shame because I last visited them on August 31, 1978 (before my Horizon kit was even delivered!) and bought a 16K RAM kit and three North Star Software Exchange (NSSE) diskettes. I wasn't charged any tax because my address was out of state! It appears that the level of North Star's service decreases greatly with time!

I wanted to look at the newest Software Manual to decide if it contained enough new material to warrant my spending \$25.00 for a copy. The people I talked to told me that they are not set up to sell anything at the plant and referred me to the North Star dealer in Berkeley (I didn't have time to see what this dealer offered in the way of service). (Later on I found out that this manual hasn't been changed at all.)

I talked to two of the service people at North Star. The second one was very informative. She works in the international sales division to which I was referred by a supervisor (?) who was in the lobby at the time and heard me mention INSUA! This was part of the royal runaround I encountered during my visit. I mentioned that the dealers in Tucson were terrible, but all they could do was refer me to the Berkeley dealer.

The second woman I talked to told me a number of things I would like to pass on to our other members. The "deal" with Computerland is defunct. Each store is

independently run and the service they provided was very variable (the owner of the Tucson store refuses to carry any North Star's). The North Star Software Exchange is also no longer in business, which is a shame.

I disagree about the value of the dealer network when it comes to handling user support. Maybe they can handle the simple questions, which saves North Star from having to answer the "dumb" questions. I called North Star last year about a method to allow more than one user to access a file at the same time while using TSS/C and North Star's CP/M. The TSS/C manual hints at a byte which can be changed to allow this, but doesn't give any hints about where this byte is located! My boss told me to tell them that I was with a dealership. I was working out of town and didn't get to receive the answer later on (he took the call). We ended up having to ignore this problem which is a stupid way to run business software. There has to be a better way to get help than this!

The only qualm I have when I recommend North Star's hardware is with the user support. I try to provide this support, which the dealers apparently can't supply and which North Star does a poor job at. Even the limited amount of software available doesn't faze me as long as CP/M can be used. I think that North Star has a much longer way to go before it provides acceptable user support than the Compass articles hinted at.

Now North Star is admitting to having financial troubles (Compass, Vol.III, No.4, p.4) and is coming out with an IBM-compatible. I hope it blows the competition's socks off! I do wonder what will happen to the older models which are still among the best computers available even today!

#

Commentary

By Steve Noll

On Compass, Vol III, no. 4

Page 15-17: Really, aren't those CP/M Users Group disks pathetic? I have bought several myself and have found them only a little better than totally useless. Yes, I know they aren't out to make money and the idea of the whole thing is great, but it is all ruined by the lack of documentation, etc. If they could only cut their library by 50% and concentrate on making the remainder useable! I probably will never buy another CPMUG disk.

Page 18-19: This is the absolute best computer-related article I have ever read in my life. Is Mr. Coudal a professional writer? If he has written any books I sure would like to know.

Other Stuff...

Sure would be nice if INSUA could work out a deal with the APCBASIC people. I would gladly buy a copy of it, but \$400? Just who are they trying to price out of the market? Too bad.

One thing that really bugs me about the INSUA is the policy not to sell the mailing list. At first glance this seems to be a good idea, privacy and all that, but there are great advantages to all if it is for sale. Who would buy it? I would think mainly companies selling NorthStar related items. Hey, I WANT to receive NorthStar related junk mail, don't you? BYTE is just too expensive to advertise in for such a small market, so you sure don't see many ads there for SCAN & REN or the stuff the Computer Shoppe sells. I can't imagine that many outfits selling TRS-80 disk drive oil would be bothering us.

I don't think that the selling of the mailing list should be a decision made by the powers at the INSUA. Why not ask the members? Maybe in the next issue? Nothing like democracy! How about on the renewal form? A box to check if you don't mind receiving ads, a box to check if you don't want to. Simple.

And then there's the money INSUA could make off the list. You could use it to...well, buy me a copy of APCBASIC!

I've written my own GBASIC program to draw machine electrical control schematics with my Advantage. Prints 'em on my NEC 8023. Does the file handling, etc. but not yet for my M hard disk. Anyway, I recently bought a copy of ImageMaker as it looked pretty neat. But what a bummer! Only works with a lousy Epson MX-100. Real foresight shown here by NorthStar. They have several modules for different plotters, but not printers. The printer stuff seems to be built into the main program, may have been compiled too, a mess to disassemble.

I wrote N* and they sent me a copy of "North Star Graphics Family Output Driver Application Note." That was nice, but it said right off that my Imagemaker rev. 1.0.0 doesn't have several files needed for developing custom drivers and I need to get them from a Busigraph II manual. Oh, swell. Wonder how much that would cost me. The note contains a source listing of an "Epson Screen Printer" which is very interesting, but too bad my brand new rev 1.0.0 doesn't use this file. I get the feeling that rev 1.0.0 must be sort of "developmental" and never should have left the factory. I also think (hope) maybe the next revision will be straightened out.

The Application note was sent to me by a Mr. Enger. This was in response to the ImageMaker Readers Comments form I filled out! I sat down one day and filled out several of those things with all my gripes. I even mailed them too! And whadaya know, one of 'em worked! I sometimes wonder if I am the first guy to send one of those in. I think it is a good idea to put a blank page at the start of your NorthStar binder(s) to jot down complaints, compliments, and questions as you run across them. After several have collected, fill out that Reader Comment page.

Mr Enger told me that the application note is normally available through my dealer and to check with my dealer for future updates or if I have any questions. Sure. My dealer has never received any updates of any kind from N* and he bought a copy of ImageMaker just before I did and is in the exact same boat.

MagicBind FollowUp

Compass, Vol. III, no. 4, pp. 12-14, contained a review of MagicBind, written by your editor. Here are some responses:

INSUA:

I am sure sold against the Diablo 630 and MagicBind. Felt like I was on a rollercoaster reading that print! ... "Typewritten text has never been other than ugly in appearance." Oh good grief, take a hike. Nice justified print sure looks better, but really... Can't say I was all that impressed by the program's features.

Steve Noll
Ventura, CA

INSUA:

Based on your print samples in Compass (and the proofs you sent me) I am going to buy a Diablo 630 printer.... I believe you must have the ECS model....

Joe Maguire
Anchorage, AK

INSUA:

Re: MagicBind review: Two things bother me about the printing: the base line is wavy, and there is too much squeezing up for justification purposes. The wavy base line may be the fault of the printer itself, not the program. The squeezing has to be done by the program. The review is pointing out the improved appearance of proportionally spaced letters; that's fine, I like it. The review says nothing about justification (adding or removing space to make the right margin straight). There are many examples in the review where letters touch each other, e.g., the word "happen", p. 12, col. 1, next to last line. This is too close. In real typesetting I believe this can be controlled. I hope it can be done in your system also.

One solution is to go to ragged right margins, which has become fashionable in many publications in the last two or three years, but I don't like it. For computer people it's a distraction--if there is a

large space left at the end of the line, I am always looking at the next word to see why it was too long to go on to the previous line.

I am amazed at your patience with the formatting of the second column. I think I would rather cut and paste it.

(I didn't write this on my computer, which is in the shop for repairs. The store where I bought it, The Computer Workshop, Rockville, MD, no longer services North Star, and they did not notify me. Fortunately, I did call them up ahead of time and found that Sorbus is providing service in this area.)

Nora M. Taylor
Bethesda, MD

INSUA:

The use of MagicBind does make the Compass look a lot better. After you get the "bugs" out it should serve quite well. Have you noticed that MagicBind has a problem with slashes (/)? Look at p. 23 where no less than seven division slashes were converted to spaces! That sure made a mess of my examples. Also, what happened to line 1010 on p.20? I had to laugh at that one. Keep trying! Also, MagicBind text makes the regular program listings look overly large. I think that program listings SHOULD NOT be set in proportional type due to the problems this causes when counting spaces (for formatting strings, etc.)

Saul H. Levy:
Tucson, AZ

!!!

Editor's response:

It is important to know what problems are attributable to MagicBind, what to the printer, and what to other elements in the chain of events that leads to the printed text.

I insist on the unredeemed ugliness of standard type-written text. MagicBind provides some solutions. We have solved the problem of the out-of-whack line on the top of the second column with the "dummy-line" routine--compare this issue to the previous one! MagicBind provides justification not only by inserting spaced between words, but by inserting spaces between characters within words. I believe this is an erroneous solution to the problem of justification, and I would recommend to MagicBind that it be eliminated, or optionally suppressed. Nora Taylor noticed that the characters sometimes touched each other. In a misguided moment, I set character width to 9; in the present issue, character width has been set to 10, for better results.

A poorly aligned printer can create uneven baselines; in the case of the last issue, however, the problem was with the printwheel, which unfortunately had a low "m", a high "p", etc. The wavy lines were thus caused by neither the Diablo 630 nor MagicBind. Unfortunately, my 630 is not an ECS (extended character set, with additional, non-standard characters). Rather, I had purchased a 96-character metalized printwheel with characters which did not match the normal 96-character plastic wheel. I still do not have the proper 96-character metalized wheel, and thus must enter certain punctuation marks by hand. I had meant to put in the back-slashes, but forgot. Also, I had forgotten to substitute the caret for the dagger several times in the review.

Our graphics advisor insists that it is a very great mistake on our part to retain right justification; he much prefers ragged right. We may eventually go to ragged right, but suffer perhaps under a false sense of grandeur at the thought of being able to justify text automatically.

Mr. Levy is correct in stating that proportional spacing should not be used for program listings. An option in MagicBind allows proportional spacing to be turned off; even so, the proportional letters give the listings an odd, probably unacceptable appearance. The only problem for the editor is that listings cannot be printed along with the text, but must be printed separately and pasted in by hand--a nasty job.

MORE ON APC BASIC

INSUA:

I was pleased to see the recent review and discussion of APCBASIC. I have owned and used a Horizon for nearly four years. One reason for purchasing it was the rich dialect of BASIC. Using North Star BASIC, I wrote a project management system which used actual resource availability to do project scheduling. I had become disenchanted with main-frame critical path calculations which did not do this, and hence were often misleading.

Unfortunately, the eight-bit, 56K version was commercially inadequate. The project size was limited, frequent disk operations made processing time too slow, and my code could not be protected. Compiling might have increased protection, but could not improve on size and speed.

I had heard of APCBASIC, but had not considered seriously until it was suggested to me by Mr. Allen Ashley, as a way to move into the sixteen-bit world. Last summer, I converted all of my programs to APCBASIC. This conversion was 99% electronic. An Horizon and an IBM PC were linked via ASCOM to transfer my code. Since I was transferring a CP/M version, some editing was required to remove disk re-sets which are not required by MS DOS. My work after that entailed enhancements to use the greater memory, and modifications available only with APCBASIC to speed up operations. The resulting programs have been tested on the COMPAQ, the Seequa CHAMELEON, and the WANG PC, as well as on the IBM PC.

The point of all this is simple. Any member who is considering a sixteen-bit upgrade need not worry about re-keying all of his programs. There is a version of APCBASIC, which will greatly extend his horizons; perhaps even his Horizon.

Very truly yours,
R. Brodie, III, President
Azimuth Group, LTD
11711 Farmland Drive
Rockville, MD 20852

Whither Micro Design?

Last month, your editor sent the following message to Alfred S. Llorens, apropos of Micro Design Associate's FC-100 controller board:

Dear Mr. Llorens:

I wonder whether you could give me (and Compass) some assistance with respect to the FC-100 controller from Micro Design? Partly on your recommendation, I have purchased the board, with the associated software. Yet it is perfectly useless to me, in part because the documentation is absolutely the worst I have ever encountered in my life. Nowhere is the configuration properly described; nowhere is there a step-by-step description of how to proceed. I count myself somewhat intelligent on the subject, having recently succeeded in installing a rather out-of-the way hard disk drive with a reconfigured CP/M; yet I have scarcely any idea of how to get started with this installment.

I might add that however stupid I myself might be, Clyde Steiner is a gifted electronics/computer tinkerer, and he is as much in the dark as I am.

Even when I blindly follow the model in the disk documentation, I fail. I have used DDT to "reconfigure" Micro Design's own sysgen program, but it crashes the instant I give a carriage return after designating the destination disk. It also crashes when I don't reconfigure it. I am also startled that it is necessary to purchase M80 and L80 to install this program--some warning should have been given in the advertisement that further expenses would be involved.

I suppose part of the trouble is that, since the programs came to me on an 8" disk, it isn't meant in that format for a N* Horizon. Yet how is one to get started?

I thank you profusely in advance!

Alan H. Nelson, Editor

Following is an excerpt from Mr. Llorens's reply. Obviously, neither he nor Compass can vouch for the details; however, let the prospective buyer beware!

Dear Mr. Nelson:

I am sorry you are having trouble with your FC-100 controller from Micro Design. I questioned Micro Design concerning the documentation, and they assured me it was just because of my ignorance that I couldn't understand it. Thus far, I am glad that others are having the same problem I had with it.

Micro Design Associates no longer exists as a company; it is undergoing bankruptcy procedures. However, the controller board may be picked up by other individuals and marketed. However, this is still in a state of flux, and the final outcome is as yet unknown.

I would have the following suggestions in regard to your FC-100 controller board. People who have dynamic memory are unable to use this board. It requires a static memory system. It also requires a boot ROM that is to be placed on either memory board, or, in the case of North Star, there is a slot on the processor board. The program for this boot ROM is the monitor program on the disk that was supplied to you. I don't think you'll have to reconfigure your BIOS, as all the intake and output ports are set for standard North Star. However, if you have changed any of the ports, then you will have to redesign your BIOS and reassemble it with the M80 program. P.S. Don't put your ROM on the FC-100 controller board. This apparently only takes a shadowable ROM.

Editor's further note:

It seems typical of poor marketing procedures that the requirement of a static memory system is nowhere mentioned in the advertisement or documentation. Nor is it clear whether the board can automatically read IBM-PC, Kaypro, Osborne, etc. format, or whether changes have to be made in software or in the PROM to enable such readings. The documentation begins as follows: "Just by virtue of you the reader being able to read this file, I will assume ..." This contradicts the first rule of documentation, which is: ASSUME NOTHING!!!

Odds and Ends

COMPASS ADVERTISING RATES

CORRECTION

Joe Maguire reports that Listing 2 of his article "Two-Column Update," Compass, Vol. III, no. 4, p. 28, contains a slight error. The first line, where the dump address is given, should read:

>DH 266B-266D

Morrow Hard Disk

By Richard M. Johnson

I have just finished (if there is any such thing) installing a Morrow 16 Mbyte hard disk on my Horizon and doing corresponding software and patches. For those of you who have purchased this drive, I must note that the Morrow manual is WRONG in one respect. When formatting your drive you must enter the number of intersector bytes as the **complement** of that number, NOT as the number itself. I spent two days wondering why I could only get 19 sectors per track instead of 32 (256 bytes/sector) until it dawned on me that their controller was upcounting to zero. The end result was that the controller was wrapping around and over-writing the first sectors without generating any error message.

For Sale!
Still some left!
North Star DD disk controller

I still have a few boards left. 90 day warranty. DOS 2.1.1, BASIC Utilities. With documentation. \$290.00 includes shipping by PP.

I also have one SIO kit (second serial port interface) and one PIO kit (parallel interface), \$20.00 ea. Includes shipping.

Joe Maguire
2321 Foxhall Drive
Anchorage AK, 99504
(907) 333-0897

The Board of Directors of INSUA has agreed that Compass should begin soliciting advertising, especially of North Star compatible hardware, software, publications, etc.

Rates, until further notice, are as follows:

Full page: \$225.00
Half page: 125.00
1/4 page: 90.00

Current circulation of Compass is about 1,500 copies six times per year. The current average print-run is 2,500 copies. All correspondence should be directed to INSUA, P.O. BOX 2910, FAIRFIELD, CA 94533

BACK ISSUES

The INSUA Board of Directors has agreed to make back issues of Compass available at the following price per Volume of 4 issues. This offer is good through 1 June 1984.

One volume: \$20.00
Two volumes: \$30.00
Three volumes: \$40.00

The volumes published to date are:

Volume I----1981
Volume II---1982
Volume III--1983

Please specify volume(s). All orders will include a free copy of the special 1983 tabloid Compass.

As noted on the inside front cover, a full index of Volumes I-III will be published in the 1984 Special Issue, scheduled for mailing shortly after 1 April 1984.

Serial



Parallel



Printers! Plotters! Punches! Robots!

Convert What You Have To What You Want!

- ★ RS232 Serial
- ★ 8 Baud Rates
- ★ Latched Outputs
- ★ Centronics Parallel
- ★ Handshake Signals
- ★ Compact 3¼ × 4¾ × 1½

only **\$89.95**

Connector Option \$10.00
CA Residents 6% tax
UPS Shipping \$3.00



1501-B Pine Street
Post Office Box 2535
Oxnard, California 93030

No longer will your peripheral choices be limited by the type of port you have available! Our new High Performance 700 Series Converters provide the missing link. Based on the latest in CMOS technology, these units feature full baud rate selection to 19.2K, with handshake signals to maximize transfer efficiency. Detailed documentation allows simplified installation. Order the Model 770 (Ser/Par) or Model 775 (Par/Ser) Today!

**Buffer Products
Coming Soon!**



CALL (805) 487-1665 or 487-1666 For *FAST* Delivery



Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203

TATE 2766
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

Postmaster: Address correction requested.
Please enter key number on form 3579.

Karl Sterne on the HORIZON

Karl Sterne, HORIZON product manager for North Star, addressed INSUA's annual meeting in San Francisco on March 24, 1984. Here are his answers to "the most frequently asked questions about North Star computers":

Q: How much longer will North Star continue to build and support the Horizon?

A: Our planning goes out for 3 years, and we foresee no halt (or even decline) in Horizon sales; we can't forecast beyond that, but expect to support the Horizon for several additional years. Of course, as we introduce improved products we will phase out obsolescent hardware such as 14-inch hard disks and single-sided floppies, and older software such as HDOS and TSS/C.

Q: Will you continue to improve the Horizon?

A: Yes. We're presently working on larger capacity hard disks and the next release of TurboDOS. We're also evaluating additional products such as asynchronous communications.

Q: Will North Star sell 16-bit applications?

A: We have no plans to sell them through North Star, as we do have two programs that help you buy directly from software suppliers: the Compatible Software Directory, which lists applications that run on North Star computers; and the Blue Binder Program, in which the software is warranted to work properly on our computers.

Q: Will you put UNIX, 68000, 100 MB hard disks, etc. on the Horizon?

A: No, but many other companies have done so, and North Star is setting up a validation facility to test these products.

Q: Will MS-DOS run on the Horizon?

A: Watch for an announcement at N.C.C.

Q: Will Total Business Solutions financial packages that now run on TSS/A run on TurboDOS?

A: Eventually. Until they do, North Star will continue to supply TSS/A software and hardware.

Q: Will the Horizon be added to NorthNet?

A: No. NorthNet's superior performance stems from its being hardware dependent on the Advantage. A Horizon gateway may be developed later.

Q: When will the Advantage get a detachable keyboard?

A: Some time after the IBM Selectric gets one.

Q: Will the North star Dimension replace the Horizon?

A: No. Both will be true multi-user computers, but the Dimension runs 16-bit PC-DOS on the IBM bus, while the Horizon runs TurboDOS (8-bit plus 16-bit CP/M) on the S-100 bus. North Star considers these as separate markets.

Q: When will the Dimension start shipping?

A: Next month.

Q: When will North Star go public?

A: We are looking forward to going public, as soon as the financial marketplace is favorable--hopefully in 1985.

#

The Compass

The Compass is published every two months by INSUA, the Interational North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, Compass, P.O. Box 2910, Fairfield, CA 94533.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

INSUA INDEX

The Index to Compass printed in this issue is based on a draft index originally created by Walter B. Rice. Articles with listings in BASIC, assembly, or other language, are denoted by a "Y" in the "LST" column. The 1983 "Special Issue" (in tabloid format) is arbitrarily designated as volume 2.5. The Editor solicits corrections and improvements to this Index.

INSUA members should take note of the articles in the various volumes as a guide to the purchase of back issues. Please see special offer for back issues on the renewal form accompanying this issue.

A complete listing of disks in the INSUA library is planned for the next issue of Compass.

What is INSUA?

INSUA is delighted with the opportunity to present itself, through this special issue, to registered owners and users of North Star computers.

INSUA, the International North Star Users' Association, was formed in 1980 by users of North Star and North Star-compatible computer equipment. The main purpose of INSUA is to assist users of North Star equipment to make the best use of that equipment by exchanging information, answering one another's questions, reporting problems or successes, sharing public domain software, and reviewing commercial software and peripherals.

The two principal instruments for achieving these purposes are Compass, the official newsletter of INSUA, and the INSUA software library.

From a handfull of members in 1980, INSUA has grown to over fifteen hundred members as of January 1984. INSUA conducts one formal meeting of members each year, at the San Francisco Computer Faire, usually in the late winter or early spring. Members who cannot attend the meeting in person can vote by mail ballot. The routine business of INSUA is conducted by the Board of Directors, who meet monthly in the San Francisco Bay Area.

Membership in INSUA, currently \$20.00 per year, includes a subscription to Compass, and, with timely registration, a free disk. Back issues of Compass are available and may be purchased. Disks from the library are sold at the cost of the media plus a handling charge. No charge is made for the programs or documentation per se.

All members of INSUA are invited to contribute articles or questions to Compass. Material longer than a few paragraphs and all programs should be submitted both in hardcopy and on North Star-readable disks (5 1/4", DOS or CP/M). All disks will be returned to the owners after the files have been copied for INSUA's use.

The Board of Directors feel that INSUA's value to members is directly proportional to the size of the membership: a larger membership means more experience and an increased capacity to discover solutions to problems. The Board therefore invites all North Star users to join INSUA. A special invitation is extended to users of the Advantage, which has now joined the Horizon as the delight of professional and hobbyist alike. We also look forward to participation by owners of the new Dimension by North Star.

North Star Answers

By Bob Cowart

Some of the Most Frequently Asked Questions... Addressed by the INSUA answer man...

Q: Some time ago I received your Disk # 1023 (DOS/BASIC 2.1.1. DQ System Disk). I have a N* Horizon, Televideo 910 terminal, and an Okidata 82A printer. I am now using DOS 5.2 with 8 digit precision BASIC. Will I need to make any changes to the new DOS in order to run it on my system? Also, every time I try to load BASIC, I get the following message: "WARNING: Memory configuration error, high mem set at 49151 NS BASIC on Horizon. Version 5.5.0".

RC: All North Star DOS disks come from the factory preconfigured for a "standard" Horizon. That is, the terminal is expected to be connected to the left serial port, and the printer to either the right serial or the parallel port. Baud rates are determined by hardware jumpers on the motherboard (the large circuit board inside the computer) so there is no need to worry about that. However, there is one major difference between the older DOS's and the new one. There is now a "sysgen" program which the user should run. This program lets DOS know how much memory your system has, and should prevent the warning message you receive when loading BASIC. Simply boot the new DOS, type "go sysgen" and answer the messages correctly. Unfortunately, there is no new documentation from North Star for DOS 2.1.1. North Star is very busy with new product development, and DOS had to take a back seat. It's amazing a new one was released at all.

? ? ?

Q: I have had no success in connecting a serial printer to my Horizon and Advantage. Do you have any suggestions?

RC: First off, be thankful you have a serial printer. Connecting a parallel

printer to a Horizon can be a real bear. Secondly, keep in mind that printer-to-computer interfacing is often complicated enough that people can make a living doing it on a professional basis. So, with that in mind, take a deep breath and get ready for a bit of printer theory.

The connections you need to wire up to your printer use the following pins on the "Right serial port" on the Horizon. The pin numbers and functions are the same on the Advantage. If you look very closely inside either a male or female connector, you can see the numbering.

Pin	Name	Function
1.	ground	chassis ground.
2.	txd	data from printer to computer
3.	rx	data from computer to printer
7.	ground	signal ground
20.	dtr	data terminal ready Indicates when printer is ready to receive more data

Now for some explanations.

One thing to keep in mind is that these names (i.e. rxd and txd) were created back in the days of teletypes, and refer to the world from the printer's point of view rather than from the computer's.

Secondly, neither CP/M nor DOS pays any attention to pin 2 (txd). Originally, the purpose of this pin was to allow a terminal or printer to send information back to the computer, but this is rarely used with microcomputers. For example, many better quality printers can send

messages to the computer on pin two, indicating whether they are ready to receive more data. Since computers can send information faster than printers can print it, there has to be some kind of "handshaking" arrangement to coordinate the two. The X-on/X-off and ETX/ACK protocols use pin two for this purpose, but the operating system, or program doing the printing, has to keep "listening" to see whether the printer has sent a stop or start message. This requires more software, and besides, not all printers can generate these special codes.

Due to these considerations, it is easier to use pin 20 (DTR) for handshaking. Almost all serial printers have one pin on their serial connector which will go from a positive voltage (about +10v) to a negative voltage (about -10v) when the printer is "busy" (cannot accept any new data). This is called the "hardware" handshaking line or DTR signal, and is usually on pin 6, 11, or 20 on the printer. Refer to your printer manual, looking for something like "...this pin changes from high to low when the buffer is full or other error condition exists..." Then make sure this pin gets connected to pin 20 on the North Star end.

As for pin 7, this is the signal ground, and, though not always necessary, is a good idea to connect. Then, make sure your baud rates are set correctly on both the printer and computer. (Baud rate on the Horizon is hardware determined by the right serial port header. The Advantage uses software baud rate setting.) Test your hookup from the operating system prompt. In CP/M type P then DIR. In DOS or GDOS it's LI#1 (or #2 if your SIO is in slot 2. etc.) If you get anything at all, then congratulate yourself. If what printed looks like garbage, your baud rates are probably set wrong. Keep trying!

? ? ?

Q: I use a Horizon, but a friend has an Advantage. How compatible are the two machines? Can we trade disks?

RC: In terms of disk format, the two machines are very compatible, with a few reservations. All Advantages use double sided, double density drives for floppy

disks. Disks used on the Advantage can be formatted double or single sided, but not single density. Nor will the Advantage read single density disks created on a Horizon. Many Horizons can read and write single or double density, but since some Horizons only have single sided drives, obviously these machines cannot handle double sided disks. The older single-density Horizons are thus incompatible with the Advantage. But if the Horizon you use is double density, you should be able to swap disks with an Advantage owner by following these rules:

- 1) If you are creating a disk on the Horizon to be used by an Advantage, just make sure it is double and not single density.
- 2) If you are creating a disk on an Advantage to be used by a double-density Horizon, then single sided is a sure bet, and double sided will work too if the Horizon has double sided drives.

? ? ?

Q: I recently bought a CP/M program which included an installation program. However, the North Star Advantage was not listed as one of the computers/terminals supported. What can I do?

RC: Unfortunately, the Advantage does not exactly emulate any popular terminal. However, many programs will run on it if you specify the Lear Seigler ADM-3a. If you still have trouble, this probably is due to the clear screen code. (The command which erases the screen differs from the ADM3a.) The Advantage requires either a one or a two byte command to clear the screen. In hexadecimal notation, the bytes are one of the following: either

1E 0F (hex) (home the cursor, clear to the end of screen)

or a single byte:

04 (hex) (video reset-turns off smooth scrolling, key-click, inverse video, etc.)

So, if your screen doesn't clear after installing for the ADM, run the install program again and change just the clear screen codes. You may have to consult the manual supplied with your program for more instructions.

? ? ?

Q: The Advantage graphics printing routine doesn't work with my Okidata printer. Why not? Is there another program which will work?

RC: The graphics dump routine was written for the Epson MX-100 or MX-80 with Graphtrax. You will have to rewrite it for use with the Oki or other printer. I can send you a copy of something similar to the source code for the graphics dump if you include \$1.00 for postage and copying. Better yet, if graphics is what you need, look into the program called Imagemaker. It allows you to do all sorts of nifty drawing on your Advantage with subsequent dumps to a number of different printers and plotters. Your North Star dealer should have more information.

? ? ?

Q: I'm confused by all the operating systems which North Star supplies for their machines. Can you give me a quick rundown on each one? Also, which ones can be operated simultaneously or at least stored on the same hard disk and used separately?

RC: I'll break this one down by machine.

Advantage

The Advantage will run the following operating systems: GDOS, GHDOS, GCP/M, MS-DOS and HDSO. The "G" when it occurs stands for "graphics." MS-DOS requires an 8/16 plug-in board from North Star along with GCP/M revision 1.2.0. (That's North Star's number, not Digital Research's.)

Two versions of MS-DOS have been released for the Advantage so far: 1.0.0. and 1.1.0. The latter is required for reading double sided IBM-PC disks.

Version 1.0.0 only reads single sided IBM disks.

HDSO is the operating system needed to run the TBS (Total Business Solution, previously called ASP) software from BSG (Basic Software Group) in Canada. HDSO is only required for hard-disk use of TBS on Advantage.

Horizon

Now for the Horizon. Of course it all started with North Star DOS (the first disk operating system available for microcomputers). Then came CP/M, HDOS (the hard-disk version on DOS) and the Time Sharing Systems - TSS/A and TSS/C. TSS/A runs the TBS software in a multi-user environment, and TSS/C allows HDOS and CP/M to be run simultaneously in a multi-user environment. However, there are now some significant changes in the offing.

Firstly, TSS/C is being dropped in favor of TurboDOS. (Since an in-depth description of the TurboDOS system is slated for a future issue of Compass, I won't cover it here.) The rationale behind this decision is based on the need to improve the performance of the Horizon multi-user systems as well as to offer the capabilities of record-locking and of 16 bit microprocessors. Using multiple processor boards (both 8 and 16 bit), TurboDOS will emulate MP/M-80 and MP/M-86. Since TurboDOS's performance is considerably superior to that of TSS/C, the latter was dropped. The only liability incurred from this decision is that HDOS and CP/M programs can no longer be run simultaneously by different users. However, if you have enough hard disk space, a half-way point can be reached. HDOS, single user TSS/A and TurboDOS can all coexist on the same drive, though they must be run separately.

Incidentally, North Star did not write TurboDOS. It is licensed from the originating firm--Software 2000. (I am in the process of testing a four-user system and will do my best to inform all you Horizon owners of my experiences in a future issue.)

#

Compass Index

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
	Who Are We?	1.1	01		Editor's Welcome to INSUA
	Movable DOS	1.1	01		Release 5.2 relocatable BASIC and DOS
	Identify BASIC Versions	1.1	02	Y	Program using ERASET command
	5.2 DOS Corrections	1.1	03		Clip Tip: Bugs in earlier versions corrected
	Clock Subroutine	1.1	04	Y	Clip Tip
P.	Command	1.1	04	Y	Gives features like CP/M to N*
	MicroSoft-N* Conversion	1.1	05		MicroSoft to N* BASIC conversion
	Electric Pencil	1.1	06		Clip Tip: Fixes for two bugs
B.	N* go CP/M = CP/M go N*	1.1	07		InfoSoft & Matchmaker conversion programs
	Bug	1.1	08		Bug in 4.0 and 5.0 N* BASIC
	BASIC Token Codes	1.1	09		
R.	ESP Test Program	1.1	10		Tests Clairvoyance
J.	Better Horizon I/O	1.1	12	Y	Tailors N* BASIC to improve some control functions
	Vietnamese Families	1.1	14		Computer used to reunite families
	Basic Entry Points/Flags	1.1	14		Clip Tip
R.	Key to Floppy Data	1.1	15	Y	DCOM Interface for BASIC
R.	LIBUTIL	1.1	16	Y	Index for software library (see 1.3 p.20)
	Proms Programmed	1.1	17		Possibly outdated
	DOS Library Routines&Flag	1.1	17		Clip Tip
R.	DOS Power	1.1	18		Command summary of program written by Pavel Breder
J.	Look at a New Editor	1.1	18		Review of N*BUS
J.	Hard Disk Drive!	1.1	20		N* HD-18 Hard Disk Drive
	Fix for DOS 5.1	1.1	20		Clip Tip: Fixes listing display
	CAT from within BASIC	1.1	20	Y	Clip Tip
	Communication Networks	1.1	21		BBS National Directory
	Northstar Hotline	1.1	21		North Star's PROXIMA
B.	N* BASIC Compiler	1.2	01		Ashley releases full compiler
C.	N* and CP/M	1.2	01		Official software from N*
R.	Mylist	1.2	03	Y	Data base for list management (see 1.3 p. 20)
S.	Pencil Points	1.2	04		Electric pencil customizing
R.	Sortlist	1.2	05	Y	Sort program for BASIC data file
S.	I/O Farm	1.2	06		Interface with other hardware
J.	Wild Card	1.2	07	Y	Code Searches for N* Monitor
J.	Hex Math	1.2	08	Y	Assembly program to perform Hex computations
S.	Long division	1.2	08	Y	A short long division program
R.	Julian Code	1.2	09	Y	Automatic date coding (see also p. 10)
J.	Move Solos	1.2	11		For Tech Sol owners
W.	Decimal to Binary	1.2	11	Y	Clip Tip
S.	Hex Math	1.2	12	Y	BASIC program to convert Hex/Decimal
P.	Modem	1.2	13		Leonard Garcia's program (INSUA disk #1003)
J.	Super BASIC	1.2	14	Y	Customize your N* Basic
	Changing BASIC Keywords	1.2	16	Y	Clip tips
B.	Horizon at 19,200 Baud	1.2	17	Y	Clip Tip (see also 1.3 p. 17)
L.	DISKLIST	1.2	18	Y	Combines and sorts directories (see 1.4 p. 28)
L.	Quickkey	1.2	20	Y	Patch enters routines with keystroke
C.	96 Tracks	1.2	23		96 or 48 tracks with N* Controller
K.	Cursor	1.2	23	Y	Call a Function to move cursor
E.	Lettergo	2.1	36		Wordprocessing software summary/review
B.	BASEX Compiler	1.3	03	Y	Compiler for 8080: Fast/Powerful.
B.	Assemble the code	1.3	07		SCS Assembler adapted to N*
P.	Destroyed File Recovery	1.3	09		Clip Tip

COMPASS INDEX

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
Rose	Disassemble the code	1.3	11	Y	Guide for N* disassembly
Jacobs	Fix for 5.1DQ DOS	1.3	14	Y	Allows DOS to boot properly
Banaghan	Tax Savings	1.3	15		Tax writeoffs with computer purchase
Leibson	N* Disk System	1.3	18		Transient: Octal system never issued by N*
	I/O Farm	1.3	19		Character codes (including ASCII)
	Northstar User Groups	1.3	24		
Jonas	64K for \$100	1.3	25		Convert a 16K board to 64K (see 2.1 p. 6)
Breder	List For CP/M Disks	1.3	26	Y	Machine language routine to read DIR
	Move It	1.3	28		Bob Hogg's PROM moving disk controller from E800.
	Disk Lib	1.3	29		Library News
Maguire	Freed from 35 Tracks	1.3	30	Y	Increase disk capacity to 40 tracks (see 3.1 p. 24)
	Letters to the Editor	1.3	33		Clip Tip (possibly transient)
Lynch	Used Computer Exchange	1.3	35		N* BASIC sort statement
Lind	BASIC Sort Patch	1.3	37		General ledger/balance sheet (INSUA disk #1008)
Warshauer	Micro Count II	1.4	02	Y	Text listing w/o line numbers (5.0 not 5.2!!!)
Nelson	No Numbers	1.4	03	Y	Bi-directional logic
Maguire	Smart Print Driver	1.4	07	Y	Power supply overvoltage protection
	Power Supply Overvoltage	1.4	10		N* programs in CP/M
Hernandez	N*-CP/M	1.4	11		N* BASIC program improves DOS I/O
Morgerstern	Swap I/Os	1.4	12	Y	Header manipulating w/switches (see 2.1 p 39)
Nelson	Easy Switch	1.4	14		Software review/summary.
Stek	dBASE II	1.4	17		Clip Tip: Parallel port controls external dDevices
	Use of Parallel Ports	1.4	19		Software review/summary
Leibson	WordStar	1.4	20		Software review/summary
Coudal	Secretary	1.4	24		List your directory in double column
Madox	2 Col List	1.4	26	Y	Convert your N* disk controller to I/O port addressing.
Rose	Disks on I/O Port	1.4	27		Tool for Removing S100 PC Boards. (clip tips)
Jonas	Tool for Removing S100	1.4	30		DBA (Doing Business as) CANSUG.
Coudal	Chicago UG News	1.4	31	Y	Basic programming techniques.
	Find/Change Variables	1.4	33		The parallel output port (see 2.1 p. 11)
Liebson	I/O Farm	1.4	36		Mass General Hospital utility system language.
	MUMPS	1.4	39	Y	Plotting planets coordinates.
Beaver	Planets	2.1	02		See 1.3 p. 25
Jonas	64K Update	2.1	06	Y	Wordstar patch creates "special keys"
Leibson	I/O Farm	2.1	07	Y	Enhancement to CP/M
Steiner	CP/M Power Is Here	2.1	12		Expanding RAM w/Expandoram II.
Lambert	Better Memory	2.1	15		Review of CP/M footnoting program
Nelson	Footnotes	2.1	17	Y	Assembly programs--disk controller & port addressing
Rose	Inside the Prom	2.1	20		Hardware review: 80 cols for SOL
Steiner	SOL Moves Up	2.1	25		CANSUG report from the windy city
Coudal	Chicago UG News	2.1	27	Y	Caloric program listing
Armstrong	Calories	2.1	29		Concerning 40 tracks
	Letter to the Editor	2.1	32	Y	Programming \$\$\$ values
Owen	Non-Cents	2.1	33		More baud-rate switches
Bryant	Easy-Switch II	2.1	39		Still more baud-rate switches
Nelson	Easy-Switch III	2.1	40		Warning on the Horizon parallel port
Yoshida	Epson	2.1	40		Warning against confusing WR and RD
Jonas	How Not To	2.1	41	Y	"DIRALPHA" assists N* DOS file handling
Coudal	DIR CLEANER	2.1	42	Y	Making BASIC input routines more friendly
Chadwick	User Friendly Input	2.2	03	Y	Diablo 1620 baud rate switch from 300 to 1200
Nelson	Diablo Baud Rate	2.2	06		

COMPASS INDEX

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
Nelson	A. Free for the ASCII	2.2	07		File transferring from N* DOS to CP/M
Leibson	S. Printer/Screen ECHO	2.2	12		Clip Tip: Echo Screen to Printer (see 2.3 p. 29)
Cowart	B. I/O Farm	2.2	14	Y	Creating I/O Drivers in N* BASIC
Shepherd	M. What SNEW	2.2	19		Report on the 7th West Coast Computer Faire
Steiner	C. CP/M Library Disk	2.2	23		INSUA disk #1011 (CP/M).
	Housekeeper	2.2	25		General utilities program: summary/review
	Northstar User Groups	2.2	26		Chicago, Dallas, Alabama, Knoxville
	Letters to the Editor	2.2	27		
	Bibliographies	2.2	30		Where to find N* subject articles
E. Coudal	Report: analog/digital devices	2.2	32		Report: analog/digital devices
S. Wasserman	Inventory-2	2.2	36		Review of software: stock control etc.
A. Nelson	File names	2.2	42		Hints for efficient file naming
B. Andrews	Aftercrash	2.2	43		Modified RESET routine jumps to Monitor
	Members List	2.3	01		Membership by Zipcode
	Northstar News	2.3	08		Address and Contact
J. Woolley	Wordstar w/Epson	2.3	09		Special print features of Epson w/Wordstar
J. Woolley	Wordstar Patches	2.3	12		Installing and using Wordstar w/Epson
A. Nelson	Epson on CP/M	2.3	15		Epson on CPM, using DOS to modify CP/M
A. Nelson	Installing Wordstar	2.3	17		Areas not covered by MicroPro
	Dotty WordStar	2.3	19		Clip Tip: Handling periods in first column
R. Beaver	Calendar Routines	2.3	21	Y	No. of days between dates
B. Stek	Expanding your Horizons	2.3	23		Speeding up your disk access with buffers etc.
B. Stek	"Catchum"	2.3	25		Review of PacMan-type software
	Letters to the Editor	2.3	26		
B. Abrahamsen	C on Advantage	2.3	30	Y	Programming in "C" with Advantage
	Questions Questions	2.3	31		Questions and answers about INSUA, WordStar, etc.
P. Midnight	PROXIMA	2.3	35		n* bulletin board system (415) 357-1130
Lind	Conversions	2.3	35	Y	US to metric (see 3.2 p. 29)
Lind	A Bit About Taxes	2.3	37	Y	Putting the tax tables in your N*
B. Stek	PILOT	2.3	38		An interpreter with new features: CP/M (review)
M. Sheppard	Pascal Corner	2.3	40		Introduction to JRT Pascal (transient)
M. Sheppard	Disk Library	2.3	41		INSUA disk commentary and news
M. Sheppard	Free Software!	2.3	42		Via the Modem
	5.0 Monitor for 5.2 DOS	2.3	43		
	Formfeed for Wordstar	2.3	44		
M. Sheppard	Miscellaneous	2.3	45		Clip Tip
R. Beaver	Finance	2.4	03		Fixes for Micro-count II and others
S. Leibson	I/O Farm	2.4	06		Equations to perform compound interest calculations
J. Maguire	Assembly Language Program	2.4	09	Y	Conversion of 32K RAM board to 64K
R. Cowart	Questions and Answers	2.4	15	Y	Introduction to assembly language programming
	CP/M Software Exchange	2.4	20		Tech expert for N* answers your questions
S. Levy	Letter to INSUA	2.4	27		Bulletin boards around the country
W. Lambert	Alien Equipment	2.4	30		Commentary on various topics
S. Cowart	DIRUTIL	2.4	32	Y	64K RAM Board, N* BASIC in CPM
R&P Nelson	North Star Answers	2.5	01		Directory Utility in N* BASIC
A. Nelson	Marry Your N* to a Sony?	2.5	03		Questions answered by N* experts
R. LaPedis	N* Questions and Answers	2.5	05		Interfacing a handheld Typeorder to a N*
O. Serlin	Some Experiences with N*	2.5	09		More answers: Assembly Language BASIC etc.
M. Perkel	What is Forth?	3.1	03	Y	Commentary on N* Advantage including HQ
R. Cowart	N* Answers	3.1	07		Evaluation of the computer language.
A. Nelson	Personal Pearl	3.1	10		Tech expert for N* answers your questions
					Review of management database software

COMPASS INDEX

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
J. Evens	RAM Test	3.1	14	Y	Running N* programs under CP/M
B. Andrews	Commentary	3.1	16		Various topics
A. Horner	Pascal Notes	3.1	18		Transient: JRT agreement never completed
	On/Off	3.1	19		Should disks be in or out under Pwr Up?
J. Orris	Bazic Listings	3.1	20	Y	File directory under Bazic or CP/M
S. Toy	DOS Assembler	3.1	21	Y	A modified SCS21 (appears on disk 1006)
	Clubs	3.1	22		Users groups info: Atlanta, Buffalo, Fremont CA.
	Vendors Column	3.1	23		Microstat, Compstat, OLIS
J. Maguire	40 Tracks	3.1	24	Y	Comments re: "Freed from 35 tracks" in 1.3.
Wm. Harmon	Business Applications	3.1	26		Comments from a real N* fan
D. Schaul	Nevada Edit	3.1	27	Y	Installing text editor
R. Ahlgren	Punctuation Protection	3.1	27		Clip Tip: Punctuation protection for 5.2 BASIC
A. Nelson	CP/M 1983	3.1	28		West Coast Computer Faire report
A. Nelson	Ribbon Renewer	3.1	29		Re-use your cloth ribbons (review)
	Letters	3.1	30		Editor answers questions from INSUA readers
	Phantom Null	3.1	02		The phantom strikes in DOS 5.2 release
	More Memory	3.2	02		Report on Morrow hard disk
J. Orris	Multi-Line Functions	3.2	03	Y	Advanced BASIC programming techniques
G. Edridge	BASIC to CP/M	3.2	11	Y	Converting N* BASIC to CP/M format
B. Stek	APC BASIC	3.2	13		Review of streamlined BASIC
E. Coudal	News from Chicago	3.2	18		Report from CANSUG users group
J. Schweppe	In Memoriam	3.2	20		Last report from Mountain View CA N* user group
C. Stevenson	Commentary	3.2	21		Comments on 2.4
Kluger	Go for Forty	3.2	23		In defense of 40 tracks
Rusin	Hard Disk with Advantage	3.2	24		Speeding up the Advantage HD
Matthews	Commentary	3.2	24		Hurrah for the Compass and other topics
	Analog/Digital	3.2	25		A to D interface: it has been done!
B. Andrews	N* Monitor	3.2	27		Modifying the 5.2 Monitor
T. Deaton	Faststep	3.2	28		5.2 DOS and a speedy disk drive
	Vendors' Column	3.2	29		Communications, Business programs, & utilities
	Bugs	3.2	29	Y	"Conversion" bugs from 2.3
	Letters to the Editor	3.2	31		Tech expert for N* answers your ???.
R. Cowart	N* Answers	3.2	35		Compass matters
Editor	Editor's Column	3.3	02		Special features analyzed for novices
E. Coudal	Features of N* BASIC	3.3	04	Y	Compact data storage using byte arithmetic
P. Garst	Byte that Fyle	3.3	10	Y	Phantom line, RAM boards, H-25 printer, Compass
Oliver	The Troubles I've Seen	3.3	11		RAM 17 static memory etc.
Lambert	Alien Equipment	3.3	12	Y	BASIC program for club mailing list
Morgenstern	Mail Program	3.3	15		Challenge to Forth enthusiasts
Levy	Levy on Forth	3.3	16		Is the Horizon still "current"?
S. de Jager	Over the Horizon?	3.3	18		Description and BASIC listing of a Nim variant
de Jager	The Game of Kayles	3.3	19	Y	Using POWER to recover lost text
Spelman III	Lost WordStar Text?	3.3	26		Compass and INSUA matters
Loschky	Irate Member's Column	3.3	29	Y	Program to improve listings in bazic
Applegate	Menu for 26	3.3	31		More news from Chicago (CANSUG)
Coudal	North Star Notes	3.3	32		Commentary on various matters
Levy	Levy on North Star	3.3	35		Various topics
S. Anuta	Commentary	3.3	40		More on power on/off
L. Matthews	On Again Off Again	3.3	42		Observations on various matters
J. Clyne	Commentary	3.3	42		Program for church treasurers
R. Applegate	PEKIN for Churches	3.3	44	Y	

COMPASS INDEX

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
Cowart	North Star Answers	3.3	46		Answers from former N* tech man
Speilman III	String Formatting	3.3	48	Y	Keeping numbers in columns on printout
	Vendors' Column	3.3	49		North Stand, CLEO, CompCare, Micro Designs, PH, etc.
	Clubs	3.3	51		Philly, Chicago, Boston, Ashtabula, and many more!
Beaver	Letters to the Editor	3.3	52		Announcements
Depke	From the Chairman	3.4	02		N* Announces the Dimension
	N* Plans 16-Bit System	3.4	04		N* announces upgrades for Horizon
	N* News Release	3.4	05		More news from Chicago
Coudal	News from CANSUG	3.4	06	Y	Non-Western alphabets on Epson 84
Eldredge	Hebrew and Greek	3.4	08		Enhancements for the DOS Monitor
Levy	Requested Monitor Fix	3.4	09		Software for proportional spacing
Nelson	MagicBind--A Review	3.4	12		Review: Catalogue system for CP/M disk libraries
Levy	Master Catalogue	3.4	15		More introductory explanations about BASIC
Coudal	Structured BASIC? Why?	3.4	18	Y	Commentary on articles in 3.3
Levy	In This Corner	3.4	20	Y	In defense of Forth
Perkel	And in This Corner	3.4	24		Update from 2.4 p. 9
Maguire	Two-Column Update	3.4	25	Y	Answers from ex N* tech expert
Cowart	North Star Answers	3.4	29	Y	Useful BASIC programs: Disklist, Pretty, and RESTTYP2
Morgenstern	Programs for N* DOS	3.4	33	Y	
	Letters to the Editor	3.4	38		
Brady	Reclaiming Filespace	3.3	36	Y	Technique for saving space on N* DOS disks
Covill	XMODEM Transfer Protocol	4.1	2		Description of standard MODEM7 protocol
Maguire	The Phantom Signal	4.1	5		History and applications of phantom line--pin 17
Nelson	Dummy Lines	4.1	7		Technique for forcing the printing of a "blank" line
Porter	Commentary	4.1	8		Dumping BASIC variables and installing quad drives
Hogan	Computer Pollution Soluti	4.1	9		Fitting a fan filter to the Horizon
Lambert	Alien Equipment	4.1	13		Ted Carnevale reviews alien S-100 boards
Lambert	WordStar Block Letters	4.1	15		Creating large block letters w/ WS and MailMerge
	Portable Advantage	4.1	16		Puget Sound Computer Service creates a portable
Maguire	Beyond 64K	4.1	17		Principles of bank-switching
Maguire	3 Laws of Bank Switching	4.1	19		Clip Tip
Morgenstern	MEMODAT	4.1	19		BASIC program to create DATA statements
	Vendors' Column	4.1	21	Y	Printer Buffer, N*CAT, AUTOTYPE editor
Appleby	Appleby on TurboDOS	4.1	23		First impressions of TurboDOS
	Advantage Graphics	4.1	24		List of vendors
Nelson	Ribbonizer Revisited	4.1	25		Followup of earlier review (3.1 pp 29-30)
Swenson	Quic-n-easy	4.1	26		Short review of program to assist in writing programs
Levy	User's Visit to North *	4.1	27		1983 visit--impressions and opinions
Noll	Commentary	4.1	28		INSUA policy, APCBASIC, and Advantage graphics
	MagicBind FollowUp	4.1	29		Portability of BASICS including N* and APC
Brodie III	More on APC BASIC	4.1	30		Problems with a promising disk controller
Llorens	Whither Micro Design	4.1	31		Correction to "Two-Column" article (3.4)
Maguire	Correction	4.1	32		
Johnson	Morrow Hard Disk	4.1	32		Installing the 16Mbyte drive

Serial



Parallel



Printers! Plotters! Punches! Robots!
**Convert What You Have
To What You Want!**

- ★ RS232 Serial
- ★ 8 Baud Rates
- ★ Latched Outputs
- ★ Centronics Parallel
- ★ Handshake Signals
- ★ Compact 3¼ × 4¾ × 1½

only **\$89.95**

Connector Option \$10.00
CA Residents 6% tax
UPS Shipping \$3.00



1501-B Pine Street
Post Office Box 2535
Oxnard, California 93030

No longer will your peripheral choices be limited by the type of port you have available! Our new High Performance 700 Series Converters provide the missing link. Based on the latest in CMOS technology, these units feature full baud rate selection to 19.2K, with handshake signals to maximize transfer efficiency. Detailed documentation allows simplified installation. Order the Model 770 (Ser/Par) or Model 775 (Par/Ser) Today!

**Buffer Products
Coming Soon!**



CALL (805) 487-1665 or 487-1666 For *FAST* Delivery



Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203

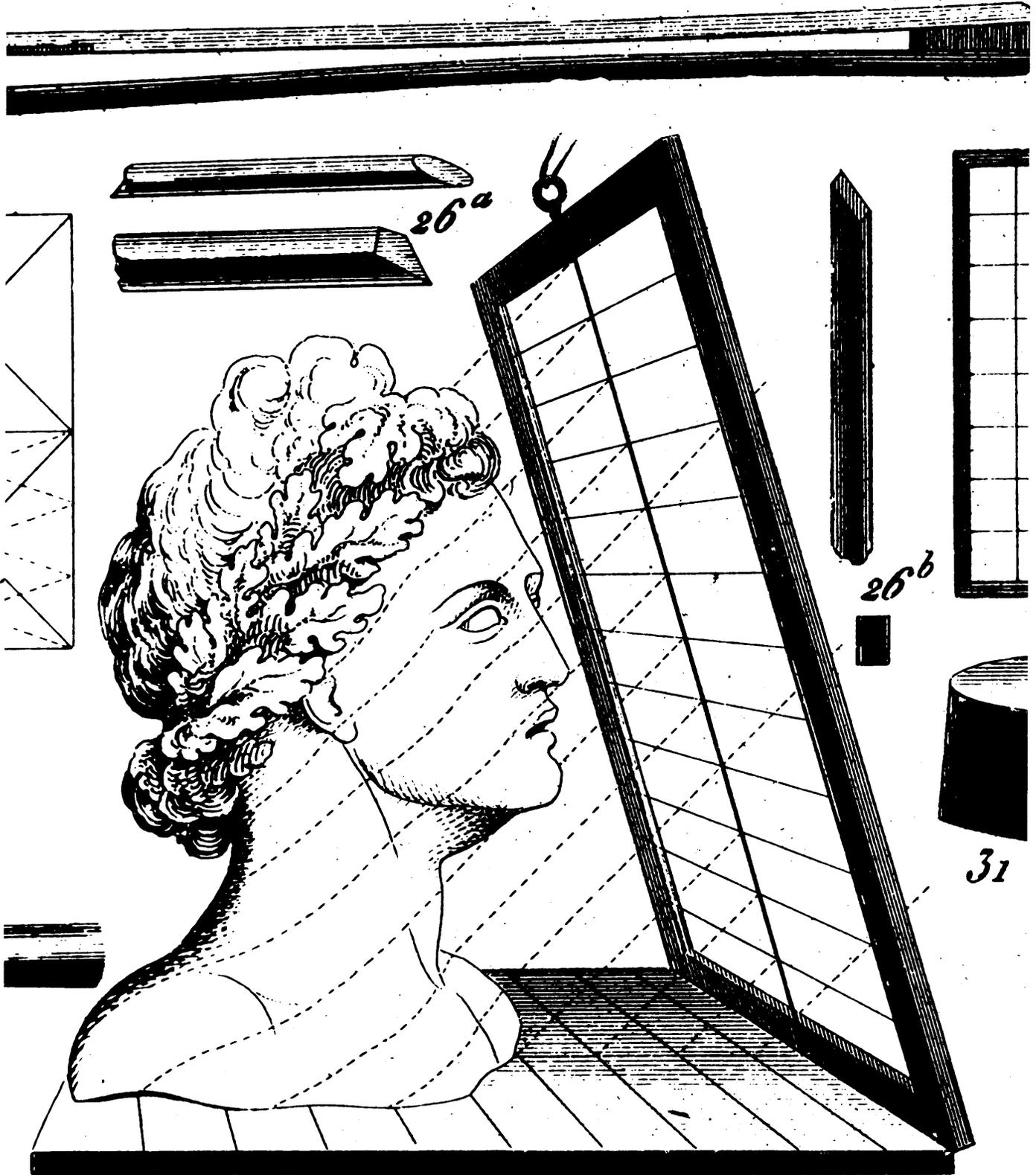
DISK OFFER EXTENDED TO JULY 1
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

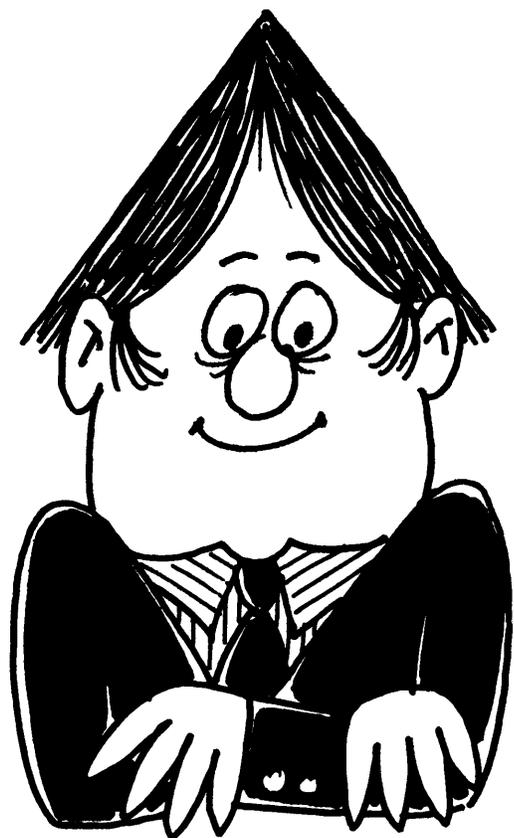
Postmaster: Address correction requested.
Please enter key number on form 3579.

The Compass

International NorthStar Users Association

Volume IV No. 2





RUN N-STAR BASIC UNDER CP/M

NOW YOU CAN RUN ALL THE NORTH STAR BASIC PROGRAMS THAT YOU KNOW AND LOVE UNDER CP/M INSTEAD OF HAVING TO SWITCH TO DOS.

YOU DON'T HAVE TO CHANGE YOUR BASIC PROGRAMS....EXCEPT THAT YOU DON'T HAVE TO MANAGE DISK SPACE...CP/M WILL DO IT FOR YOU.

for INSUA members only

INSUA has purchased the full rights for its members to the entire InfoSoft commercial software package.

You get:

NSDIR.COM - list N*DOS disks under CP/M.

NSLOAD.COM - load files from N*DOS disks to CP/M disks.

NSLIST - convert N*BASIC listing to standard ASCII text that ANY standard word processor can read.

NSENDER.COM convert standard word processor ASCII text to N*BASIC's "packed" format.

InfoSoft's manual and suggestions for other customization.

...AND MODIFICATIONS TO N*BASIC TO RUN UNDER THE CP/M OPERATING SYSTEM.

GET THE FULL PACKAGE \$20

INTERNATIONAL NORTH STAR USERS ASSOCIATION POST OFFICE BOX 2789, FAIRFIELD, CA 94533

REGULAR RETAIL \$\$\$\$\$

The Compass is published every two months by INSUA, the Interational North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, 2600 Buena Vista Way, Berkeley, CA 94708.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

- 2 Clyne B. **Installing the Morrow 16Mb**
The Morrow 16 Mb hard disk drive
- 5 Whitman J. **Vertical Horizon**
Mounting the Horizon vertically for convenience
- 6 Maguire J. **Disk Emulator: A Review**
The LS-100 Disk Simulator from Digital Research
- 7 Evert J. **File Length in DOS**
A Solution to some End-Of-File Problems
- 8 Maguire J. **The Blank PC Board Option**
Discussion of kits offered with boards but no parts
- 8 Friend A **Pizza Calculator**
Program to determine price of pizza by the square inch
- 9 Ahlgren R. **Modification of 32K Board**
Increasing available memory through board fixes
- 12 Shulkin R. **Software---controlled Switching**
Switching relays controlled from unused port
- 14 Maguire J. **Relocating the Disk Controller**
Software control of disk controller location
- 18 Ingram D. **A File Conversion Program for N***
Convert ASCII and BASIC files from DOS to CP/M
- 26 Shearer J. **A N* BASIC Pre-processor**
Write BASIC programs without linenumbers!
- 30 Levy S. **NSPRE: A Review**
Review of Jim Shearer's Pre-processor
- 36 Shearer J. **NSPRE: A Response**
Jim Shearer's response to Saul Levy's review
- 37 Maguire J. **Letters to Joe**
Advice for all letter-writers

Installing the Morrow 16 Mb

1-313-
759-6569

By Bob Clyne
11151 Edgemont
Warren, MI 48089

I had intended to buy a Winchester hard disk later this year (1983), but when I saw the Priority 1 advertisements for the Morrow M16 hard disk subsystem for \$1595 including CP/M 2.2 and MBASIC my plans to wait until late in the year began to fade into the sunset and finally after a few weeks my buying resistance and fear of buying a rather expensive turkey had dissipated and I threw caution to the wind and ordered one in early May.

It did not actually arrive until the end of June. I had several phone conversations with the salesman at Priority 1 in the weeks between and he explained that they were having trouble getting the units from Morrow but still several expected delivery dates were missed. To their credit, Priority 1 did not process the charge to my credit card until they were ready to ship.

In the meantime Sigi Kluger, in Texas, who had ordered his M16 several weeks before I did had received his unit and had it running. He had it running for about a week that is before it died. He got it back from Morrow in August, finally repaired. Needless to say this did not exactly give me a warm feeling of confidence about the units.

I have a North Star Horizon originally purchased as a kit in 1977 with a North Star Z80 processor board and a North Star 16K dynamic RAM board. To that I had added, through the years, a North Star double density disk controller board, an Industrial Micro Systems 32K static RAM board, a Seattle Computers 16K 8/16 static RAM board and a Potomac Micro-Magic MM-103 modem board. All of this was built before the IEEE-696 standard was proposed. In addition I had installed a modification to phantom out the memory over the disk controller PROM whenever it was addressed. North Star steadfastly recommended against this modification but it worked flawlessly and made several more K of memory available.

Another modification allowed my serial I/O ports on the Horizon to operate at 19,200 baud vs the 9600 baud maximum with the stock setup. When plugging a new board into a conglomeration like this one always hopes it will work right away but only the most incorrigible optomist actually expects it to. I am sometimes an optomist and frequently incorrigible but seldom both at the same time and I already knew that the North Star RAM board did not get along well with IEEE-696 equipment and the Morrow HDDMA board was advertised as complying with the IEEE standard.

The first surprise was when I could find only three stick-on rubber feet for the disk drive but I finally decided that it must have been intended that way; I remembered something about a three point suspension being less likely produce a strain than a four point suspension (same reason that a three legged stool always sits solidly while a four legged one may rock if the floor is uneven).

Then there was the documentation, which could charitably be described as meager. After studying the documentation carefully I still had a lot more questions than answers so I took the two 5.25" disks from the package and put them in the computer to see what was on them; there were also two 8" disks but since I had no way to read them they were of no help at the time. There were a few .DOC files on the 5.25" disks which answered a few questions, and opened up almost as many more. The manual for my North Star 16K RAM board had some modifications listed to operate the board with DMA devices and there was a modification for North Star 32K and 16K RAM boards on a sheet of paper included with the hard disk.

Since the North Star and Morrow modifications did not involve the same connections or reference each other, I decided to try the North Star modifications alone first, so I modified the North Star

RAM board which I had addressed as the top 16K of memory and then plugged everything into the bus, turned it on, and hit the reset. No luck, the system would not boot.

Undaunted, I shut it down, pulled the North Star RAM board and added the Morrow modification. No luck on this try either, again the system would not boot. This time I tried removing the North Star modification while leaving the Morrow modification in place. Same result, the system would not boot. I tried it with the HDDMA controller board removed and everything worked fine. I tried removing all of the modifications from the North Star RAM board; I did not really expect this to help but it was about the only configuration left to try on that board. No change, it still would not boot. I then tried addressing the various memory boards at different locations and finally found a combination that would let the system boot. The Industrial Micro Systems board was at the top, the North Star board in the middle, and the Seattle Computer board at the bottom of memory. It sure was nice to see that A> prompt.

The next obvious move was to test the memory so I ran MEMR. It ran successfully and reported no problems. That was strange since the system was so sensitive to where each memory board was addressed, but I decided to come back to that question later. I ran the INSTALL program and after a couple of tries got the command line right and the program ran successfully. A couple of more tries to get the command line right and the format program ran successfully. I now had the hard disk up and running, configured as two logical drives of about 8 megabytes each with 4K allocation blocks. This should have been the end of the story but it turned out that there were a few more chapters still to be written.

I tried running a few more memory test programs. MEMR still showed everything OK. MEMDIAG also showed everything OK, but RAMTEST3 and RAMTEST5 got through several of their test segments with no problems and then crashed the system when they got to the North Star RAM board. RAMTEST3 and 5 are really memory test programs which were

furnished with North Star DOS 5.2 which I adapted to run as transient programs under CP/M. The only difference between the two is that RAMTEST3 runs at 3000H and RAMTEST5 runs at 5000H so that all segments of memory can be tested.

Several runs of the memory test programs produced the same results and I was beginning to wonder if the failures of RAMTEST3 and 5 were significant or if the problem was something which only happened during the memory test. The answer to that came when I tried to run SWEEP38. This caused, on various tries, either the program to blow up or the system to crash, thereby establishing that there was a serious problem in the system. Another result of this was to give me a lot more respect for the ability of the RAMTEST programs to locate memory problems since they had found problems which other memory tests had missed.

The next step was to borrow a 16K Econoram IV static memory board from Jim Wolfe and try it in place of the North Star 16K RAM. The memory problems disappeared when the Econoram IV was in the system in place of the North Star board, thereby removing any remaining doubt that the North Star board was the source of the problem.

Since I intend to upgrade to CP/M+ with a banked memory system in the near future I decided that I might as well buy a new 64K static memory board now. I wanted a board which would do bank switching for the CP/M+ project and also have the capability to do 16 bit transfers to facilitate an eventual upgrade to a 16 bit processor, and I wanted IEEE 696 compliance to reduce incompatibility problems with other hardware. I rapidly narrowed the field to three boards: a Macrotech 128K static memory board, the CompuPro RAM 16, and the Fulcrum Omniram.

The Macrotech board was more than I needed and I do not know anyone who had ever used one but it seemed to have everything. It could do bank switching and map any 4K segment into any address in any bank. It had IEEE 696 compliance, could do 16 bit transfers, had extended addressing, and was fast, but the price of \$1232 was more than I could justify.

CompuPro has a good reputation for building quality static memory boards but I discovered that the RAM 16 would do 16 bit transfers but would not do bank switching and the RAM 17 would do bank switching but would not do 16 bit transfers.

That left the Omniram which would do both, supported extended addressing, and had gotten a good review in Microsystems magazine earlier this year, so I ordered one by telephone from California Digital. I had never done business with them before so I was a little worried, but the board arrived by UPS blue in less than a week which I think is excellent service on the part of both California Digital and UPS. The documentation with this board is adequate and reasonably clear but in some cases I do not think it follows the most logical order and it leaves some doubts as to whether or not the board is configured correctly. One should definitely read completely through the documentation any time you are making changes.

When I had configured the new memory board, I removed all the old memory boards and plugged in the Omniram board. The system came up but was flakey and unreliable and usually crashed after the first or second command. I finally discovered that the Omniram board would not reliably tolerate my modification to phantom out the memory overlying the disk controller PROM when the PROM was being addressed. While the Omniram does respond to the phantom line, apparently the timing produced by my modification was not within tolerance for this board. Since the modification was a field kluge there is no reason to believe it met any timing specification so this is not a fault of the memory board.

I also discovered that the Seattle Computer 8/16 memory board would not operate using the phantom line modification to phantom it out for PROM access. The Industrial Micro Systems board and the North Star 16K memory boards did not seem to mind the phantom line kluge at all though. I finally removed the phantom modification and used the provision on the Omniram board to make a 2K window for the PROM. This did result in a net 1K loss of useable memory (from

EC00H to EFFFH) but the system worked reliably.

Now that most of the serious problems were cleared up though, one more problem became obvious. There were glitches in the output to the console and to the printer which were connected to the Horizon left and right serial ports respectively. The machine uses 8251 USARTs, and in trying to eliminate the problem I substituted an Intel 8251A, which I had on hand, for the USART for the console. The computer worked fine right up until I accessed the hard disk, at which time the system locked up. I heard the hard disk head move but somewhere between there and outputting the results, the system went to never never land.

I checked the chip I had removed and discovered that it was a 9551 which is an enhanced 8251. I did not have another 9551 but I tried two other Intel 8251s with the same results as the first one. The other serial port had a NEC 8251C and substituting that in the first serial port produced the same results as the 9551, i.e. the system worked but there were occasional incorrect characters in the output. The output problems would go away if the hard disk were not accessed but that was not much of a consolation.

The modification I had installed to obtain 19,200 baud to the console was to feed a 4MHz clock signal from the bus to the input of the baud rate divider chain instead of the 2MHz clock signal it was originally using. The clock input to the divider chain is buffered by a Schmidt trigger (74LS14) but this apparently was not enough. I did not have an oscilloscope to examine the wave shape but I think the DMA must have done something to it. I finally removed my 19,200 baud modification and restored the serial output circuits to their original configuration, which cleared up both the console output and the printer output problems.

The printer baud rate remained at 9600 baud throughout and there never was any problem with errors on the input. The 9600 baud rate is the maximum the Horizon was designed for but I do miss the 19,200 baud speed to the console. The divisor to obtain the baud rate times 16 needed by the USART from 2 MHz is

approximately 6.5 for 19,200 baud so the prospects for getting back to 19.2K baud do not appear bright.

After the above changes the system worked without further problems and I later edited the IBIOS furnished by Morrow to give me three logical drives of slightly over 5 megabytes each with 2K allocation blocks (rather than the 4K blocks in the stock system) and room for 1024 directory entries for each logical drive.

The IBIOS which the INSTALL program patches into the CBIOS jump table takes 4K of memory the way I have it configured and about 3.25K using M16.PRL which configures it as 2 logical drives. The system intercepts calls to the disk system at the CBIOS jump table decides if they are for the hard disk and just sends them back to the regular CBIOS if they are for the original floppies. There are provisions in the IBIOS source file for any and all of the Morrow disk controller boards but only the code for the equipment you tell it you have is actually assembled. The M16 can be brought up very simply if you want to use the supplied configuration but if you want to have your own custom configuration all the tools are there, except for a macro assembler.

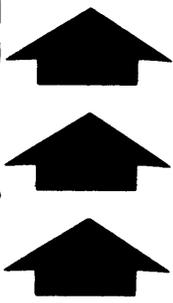
It is interesting that the M16 subsystem with the stock software saves 64K bytes on the disk for each directory but the maximum number of entries allowed would only fill 32K thereby wasting 32K of disk space. If you want to retain the 4K allocation blocks the wasted space can be recovered by editing the IBIOS to change the allocation vector in the DPB from 0FFH,0FFH to 0FFH,00H. After some time I found out that the (4)M16.PRL file causes the hard disk to be configured as four logical drives; this was not explained in the documentation.

The Morrow documentation makes no mention of landing zones for the hard disk drive heads when the unit is being shut down. The Computer Memories Inc. manual which is included says in section 2.7 Read/Write Head Landing Zone "to be added....." Not wanting to damage the unit by improper shut down procedures, I called CMI and talked to a technician named Lenny Caprino who said that they were considering special head landing

zones at one time and that they might specify some at some time in the future but that head location was unimportant for a normal shutdown. He said that he recommends the heads be positioned on track 305 (the innermost track) for shipping. When I called CMI there was only a few seconds wait to get the technician and Lenny was friendly and helpful. CMI gets an excellent rating from me in this area.

If anyone has succeeded in getting a North Star 16K RAM board to work with the HDDMA controller I would like to hear how you did it. Similarly, if anyone has any explanation for the phenomena I experienced with the baud rate and the 8251s or any ideas (short of buying a new I/O board or worse yet redesigning the Horizon mother board) to get back to 19.2K baud, I would like to discuss it with you.

#

Vertical  Horizon

By James A. Whitman
Rt. #1, Box 408
Ft. Ashby, WV 26719

I have devised a way of mounting my Horizon vertically next to my workstation. I think that it looks quite natural mounted in this fashion, and have experienced no problems whatsoever. If any INSUA members would like to do the same, I will supply the necessary hardware and instructions for \$25.00 plus shipping.

I have not developed a desk top version, but if enough people are interested, I will design one to allow the Horizon to be placed vertically on top of a desk. The price would probably be a little higher due to the additional hardware involved.

Disk Emulator: A Review

Product: LS-100 Disk Simulator
Vendor: Digital Research Computers
P.O. Box 461565,
Garland, Texas 75046
Price: \$399.00 (Kit)

Reviewer: Joe Maguire
2321 Foxhall Drive
Anchorage, AK 99504

This board emulates a floppy disk drive by using 256K of on board RAM. (DRC calls it a simulator but I think "emulator" is a more correct term.) The "LS" in the product name stands for "Lightning Speed." The "100" means that it is an S-100 board.

A disk emulator works by fooling the CPU into thinking it is a disk drive. However, instead of storing data on a magnetic surface, the DE uses conventional RAM. The RAM is usually dynamic since those chips are more widely available at the present time. The LS-100 uses 32 of the popular 4164 (64K x 1), 5V, 16 pin RAMs.

In order to trick the CPU into believing it is a disk, a DE must have special routines added to the DOS or CP/M BIOS. As long as the proper data, checksums, and status bytes are provided by the DE, the CPU doesn't know what is actually behind the "smoke screen" the DE sets up. Installing these routines may or may not be trivial depending on your DOS/BIOS and your experience with assembly language programming.

Why use a DE instead of a real disk? Speed! A DE typically operates 7 to 10 times faster than a floppy disk and about twice as fast as a hard disk. For a disk intensive program, such as WordStar, this can increase productivity tremendously.

Some special preparation is required in order to use a DE just after turning on your computer. Like any other RAM at turnon, the DE RAM contains random data when power is first applied. Therefore, the DE must be "formatted" prior to each work session. After formatting, the desired files must be copied to the DE from a real disk before the speed

advantage can be realized. After the worksession has been completed, the files on the DE must be copied back to disk or they will be lost when the computer is turned off. Much of this can be accomplished automatically by using SUBMIT files or other self-executing routines.

I installed my LS-100 in my Horizon to be used with CP/M. DRC provides an Install program (on an 8" disk) which will automatically patch your BIOS with the required jumps to externally located driver routines. These routines require about 600 bytes of additional RAM space.

The Install program must first be edited to tell it the system size and any other special information about the CCP and BDOS locations. This is required for North Star CP/M (and Lifeboat CP/M) since N* uses a non-standard offset between the BDOS and the BIOS.

After editing, the INSTALL.ASM file and a companion DRiVe.ASM file must be assembled and the resulting HEX files combined with DDT into a COM file. If you have never done this before, it may be a difficult job to get the LS-100 up and running in your system. The instruction manual is adequate but not overly clear. Examples of all the required driver routines are given so you can get an idea of what is required. Using the recommended steps outlined in the manual I had no difficulty in getting my LS-100 up and running.

A second option is to place the driver routines directly into the BIOS. This eliminates the extra RAM requirement. I elected to try this and it required me to:

1. Disassemble the BIOS. (I used the RESOURCE disassembler on INSUA disk #1014. It is an excellent disassembler.)
2. Locate the necessary routines, disk headers, parameter blocks, etc. from the source code.
3. Install the DE routines, reassemble the BIOS, and test it out.

The process sounds easy but it took me about 20 hours of hard work before I finally got the bugs out and everything running. It is definitely not a project for a novice!

Results. I found that a DE is indeed addictive! Both my wife and I frequently use WordStar and my wife is a fast typist. She is particularly annoyed at having to stop typing--often--during disk activity. Worry no more! With a DE you can type at full speed! When WS calls in various overlays, they appear almost instantly. Moving the cursor from the beginning to the end of an 80K file used to take almost a minute. Now it takes about two seconds! My impression is that the speed advantage, for me, is about 5 times. For my wife it is more like 10. It depends on the software you use and the amount of disk activity involved.

Caveats. It is important to remember to save the edited files to a disk after a work session. This takes some getting used to if you have been a confirmed save-and-resume type WS user. Save-and-resume doesn't give much insurance when using a DE. The LS-100 does have provision for a battery backup if data retention during a power failure is important.

File Length in DOS

By James Evert

With Northstar DOS a file has to be constructed to a given length. What if a write instruction is given that goes beyond that given length? The program crashes and the file is placed in such a condition as to be unuseable. The reason is that the end of the file doesn't match what the machine is trying to read.

I have found that by using the LF command of the DOS, and loading the file into RAM, at say 2000, I can then change the file by using the DS command of the Monitor. By using the DA command of the monitor the RAM can be observed and the end of the file found. By comparing this display with the printout from a program I call READ3, found in the book The Users Guide to North Star BASIC by Robert

The manual had a few typos and one serious mistake. In the description of how to use the board with a Z80 CPU, the instructions say to install a jumper at J8 for use of the refresh signal from the North Star CPU board (pin 66). The correct jumper position is J9. The schematic diagram shows this correctly.

The LS-100 is offered only in kit form. I had no difficulty with the assembly. A diagnostic program is available on the accompanying disk to aid in checkout.

Special Note: Readers who are interested in this board and who wish to try incorporating the driver routines into the BIOS, may send me a N* disk with a suitable mailer and sufficient return postage. I will place a copy of the patched BIOS on the disk and return it. My CP/M is version 2.23a from Lifeboat. This BIOS is very similar to other versions and should enable you to get your board going with minimum frustration. In order to fit the DE routines in, I had to delete some other routines such as those for single density disks and for one-drive only systems. As it stands, the patched BIOS will support SSDD and/or DSDD multiple drives with the LS-100.

#

Rogers, I was able to reconstruct the last entry so the machine could read it. An 01 end mark is also required after the last entry.

After the fix has been made, construct another file bigger than the first and save the file now in RAM into the new file on the disk by using the SF command of the DOS. Try it! You have nothing to lose.

```
10 DIM A$(1000)
20 INPUT "FILE NAME:      ",B$
25 INPUT "INPUT TERMINAL NUMBER  ",P
30 OPEN #0, B$
40 IF TYP(0)=0 THEN END
50 IF TYP(0)=1 THEN 90
60 READ #0,N
70 !#P TAB(6),N
80 GOTO 40
90 READ #0,A$
100 !#P TAB(15),A$
110 GOTO 40
```

The Blank PC Board Option

By Joe Maguire

What I didn't say in the above article on the LS-100 Disk Emulator board from Digital Research of Texas was that I purchased the board blank--that is, without parts. The only other items included with the LS-100 blank-PCB option are the instruction manual and an 8" disk with the necessary software routines.

Can you save money purchasing just the PCB, without any parts? Here is my experience.

I am an inveterate parts collector. Whenever I pass a flea market, a swap fest or a discounter's parking lot sale, I stock up. Over the years I've accumulated quite a junkbox full. Altogether, I guess I haven't spent more than a few hundred dollars on it but the list price value must be closer to several thousand.

When I see a product such as the LS-100 advertized with a blank PCB option, my mental calculator starts cranking. If I think I have most of the required parts on hand, I am tempted to buy. In the case of the LS-100, my junkbox provided only about \$20.00 worth of parts, but I still saved money. Here's the accounting:

Blank PCB with manual & disk	\$ 69.95
8203-1 RAM controller	39.95
20 Mhz Xtal	4.00
4164 RAM chips (32)	128.00
Misc ICs, heat sinks, etc.	20.00
Junkbox parts: sockets, caps, ICs, regulators, etc.	(20.00)
Total cost:	\$281.90
DR full kit price:	\$399.00
Saving:	\$117.10

I've added in the \$20.00 worth of junkbox parts even though I didn't spend it. You can see, though, that even if you purchased all the small items at your local Radio Shack store, you would still likely come out ahead.

The RAM controller IC and the Xtal I purchased from Byte advertisers. The

4164 RAM chips were bought at the West Coast Computer Faire. During the closing hours of the show on Sunday, a buying-and-selling frenzy developed which I could hardly believe. I bought the RAMs for \$4.00 each instead of the normal \$5.95.

If these are the going prices, why does a vendor charge so much more? Because he bought all his parts months or years ago when those 4164 chips, for example, were \$10.00 each. He just can't afford to lower the price until his stock is used up. In the memory market, just wait a few weeks and the price will drop ten percent!

I hope this example shows that it is still possible to get an honest bargain. Even DR's full price of \$399.00 for a 256K disk emulator is a bargain, but at less than \$300.00, it's a steal! Just compare that to the prices you see advertized for similar assembled boards. You just can't afford not to buy one and it works great in the Horizon!

P.S. For those who wonder, I am in no way connected with DR. I just like to see the word spread about a good product.

#

PIZZA Calculator

By A FRIEND OF INSUA

Have you ever gone to a PIZZA parlor and been confused by the prices? Here is a program that allows you to compare prices by the square inch. PIZZA sizes from 9 to 23 inches in diameter are shown along with cost per square inch from \$.05 to \$.15 in half cent steps.

Now, I know that this is not the most pressing concern in your life, but you would be surprised at the number of times you will find that two medium size PIZZAs are cheaper than one extra large. Enjoy. Yum! Yum!

(See p. 11)

The hardware modifications used a spare D type latch (74LS75) at location 6B pins 3, 14 and 15, a spare inverter (74LS14) at location 2C pins 10 and 11, and a spare NAND (74L00) at location 7C pins 11, 12 and 13. WARNING: the schematic on page 37 of my North Star manual erroneously shows the spare NAND at pins 4, 5 and 6 (7C); check your manual and board for the location of the spare NAND.

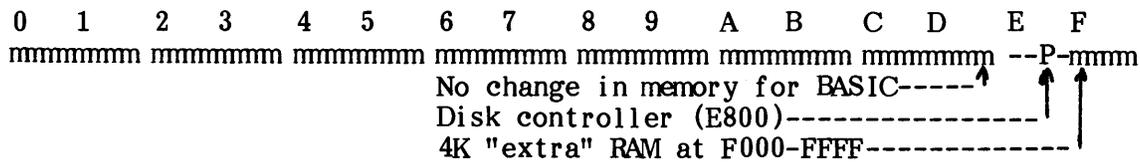
Carefully, referring to fig. 1 and the N* manual, add the following jumpers on your 32K board that has the 8K block turned off.

JUMPER NO	FROM LOCATION	PIN	TO LOCATION	PIN
1	3D	13	6B	3
2	6B	14	7C	13
3	2C	10	7C	12
4	6A	7	2C	11
5	7C	11	7A	8

Cut the runner from 6A pin 7 to 7A pin 8. Turn on the "E" or 8th switch at location 7A and return the board to the computer. You should have the additional memory shown in the "Modified Memory Map" above.

It is also possible to modify the memory board in such a way that the additional memory is in a 4K block at addresses F000 thru FFFF, as follows:

Alternate Modified Memory Map



m=1K of RAM memory
 P=1K of ROM (read only memory)

Note: This method does not extend the top of the memory for the BASIC interpreter. This modification requires changing the connection of two jumpers as shown in fig. 2.

This successful modification has passed many hours of memory testing including RAMTEST3 and RAMTEST5 of North Star's DOS 5.2.

Fig. 1 CIRCUIIT MODIFICATIONS

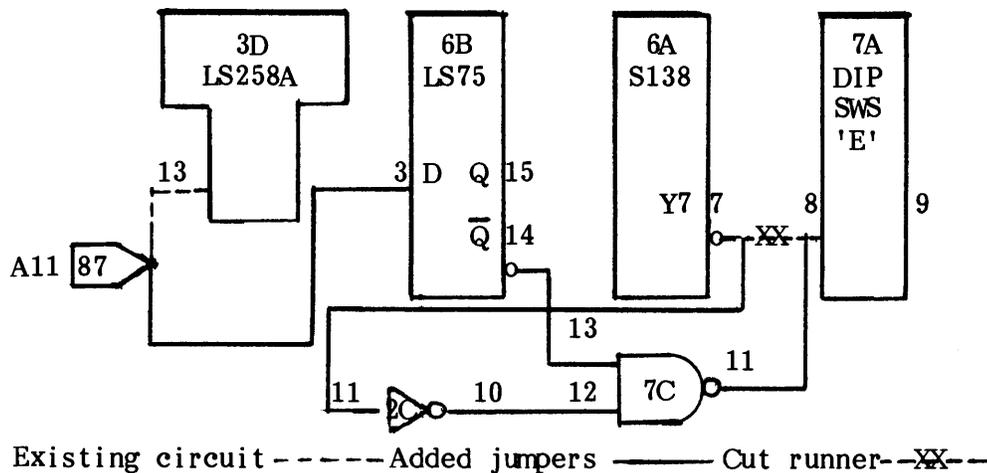
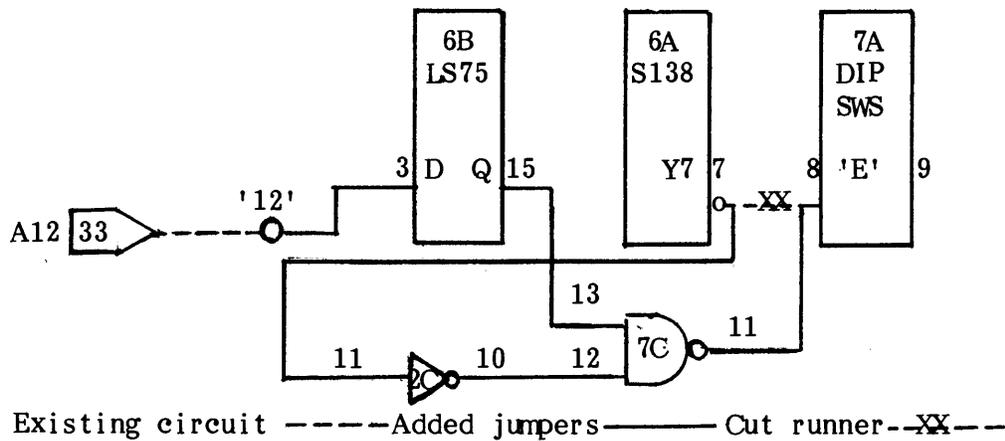


Fig. 2 ALTERNATE CIRCUIT MODIFICATIONS



```

10 REM PIZZA CALCULATOR
20 PRINT TAB(30), "PIZZA CALCULATOR"
30 PRINT
40 S$=" in."
50 PRINT "SIZE ",
60 FOR I=1 TO 7
70 READ N
80 PRINT %6I, N, S$,
90 NEXT
100 PRINT
110 RESTORE
120 GOSUB 330
130 PRINT
140 PRINT "IN^2",
150 FOR I=1 TO 7
160 READ N
170 A=(N/2)^2*3.14159
180 T(I)=A
190 PRINT %10F1, A,
200 NEXT I
210 PRINT
220 GOSUB 330
230 PRINT
240 FOR I=.05 TO .15 STEP .005
250 PRINT %4F3, I,
260 FOR J=1 TO 7
270 PRINT %10F2, I*T(J),
280 NEXT J
290 PRINT
300 NEXT
310 PRINT
320 END
330 FOR I=1 TO 76
340 PRINT "-",
350 NEXT
360 RETURN
370 DATA 9, 11, 13, 15, 17, 18, 23
READY

```



Software-controlled Switching

Roger Shulkin
SIVAD of Texas, Inc.
20214 Brondesbury Drive
Katy, TX 77450

I read with interest the question awhile back about how to obtain an 8 pole switch to change between printer and modem. (See Compass, Vol. I, no. 4; see also Vol. II, no. 1.) Perhaps your readers would be interested in using the Parallel Port and some reed relays to do the switching under software control. It's a lot easier than pulling cable plugs.

Several years ago, while I was working with my first computer, a wonderful SOL with Northstar DOS and disk drives, a very skilled technician named Bob Johnston suggested a solution to my requirement to switch Serial output between two printers. I use one printer, an NEC 5525 on Serial output, to print invoices and other reports, and my other printer, an Epson MX-80 also on Serial output, to print mailing labels for the packages. Bob's suggestion was to use the two reed relays in the SOL which were used to control cassette tapes, and which I never used, to switch the Serial port output between the two printers.

In the last few years, I have added another MX-80 to print disbursement checks, so I now use the same principle to switch the Serial output between three printers.

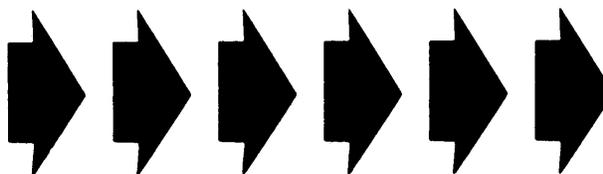
The Horizon which I use is operating under N* HDOS, Version 2.1.0. I'm on a 15 meg hard disk with HDOS at 5000 and HBASIC at 2600. But the same out commands also work under Release 5.2 and probably earlier versions also. The Parallel port can be used to send out very small TTL signals which will energize reed relays which can be used to energize larger relays to switch the Serial output between printers, modems or whatever. The principle can be expanded to turn on or off lights, alarms, or other electrical devices provided care is taken to properly isolate all voltages from the computer.

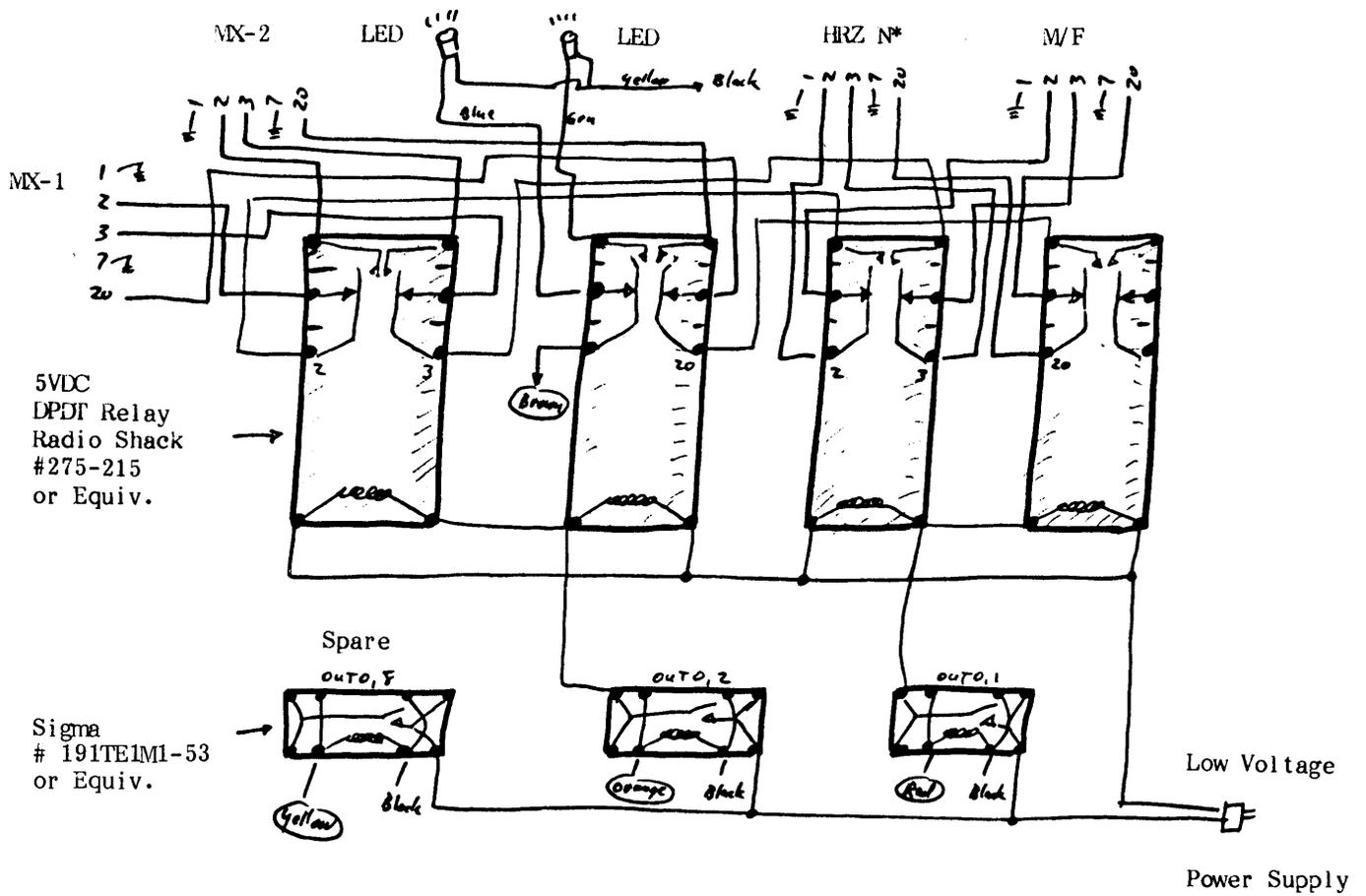
The following is a key for 'OUTO,___ commands for the Horizon:

Parallel Pin #	OUT O,___
5	1
12	2
4	4
11	8
10	16
2	32
9	64
1	128
3	Common Ground

The relays which I use are the equivalent to the SIGMA #191TE1A1-5S, which I last purchased for about \$1.50 to \$2.00.

As you can see from the above listing, it is possible to control as many as 8 separate switching functions using the OUTO, command and any combination of values listed above. To switch relays on pins 5 and 12, do OUTO,3. To switch relays on pins 12 and 4, do OUTO,6, and so on. I have enclosed a copy of my wiring schematic which is a little rough, but it works. I only use two output commands although I have extra relays for use as spares when needed. It is important to have SPDT or DPDT relays used for the actual switching, since you need a default of the main printer when all relays are off (i.e. OUTO,0). This is important when using CM/P with a word processor program to insure that you'll be able to print any work done.





N* Horz-Male	M/F Female	MX-1 Male	MX-2 Male	
1-orange	1-yellow	1-yellow	1-brown	
2-yellow	2-green	2-green	2-red	
3-green	3-blue	3-blue	3-orange	
7-violet	7-grey	7-violet	7-yellow	
20-blue	20-violet	20-grey	20-green	
1-grey	OUT0,128 (spare)			<u>KEY</u>
2-white	OUT0,32 (spare)			NEC OUT0,0 No lites
3-black	OUT0,4			MX-1 OUT0,1
4-brown	OUT0,1			OUT0,5 w/lites
5-red	OUT0,64 (spare)			
9-blue	OUT0,16 (spare)			
10-green	OUT0,8			MX-2 OUT0,2
11-yellow	OUT0,2			OUT0,7 w/lites
12-orange				

LED's Flat Notch = Neg

Relocating the Disk Controller by Software Command

By Joe Maguire

The question about relocating the North Star disk controller, which was posed by Philip Bond (Compass Vol. 3, no. 3, p. 4), got me to thinking. If one is not afraid of a little hardware modification, and some software customizing, it can be done for a lot less than buying a Micro Devices' controller board. As a plus, the method I have in mind is entirely controlled by software, which eliminates any compatibility problems.

The Requirements

Relocating the controller 4K higher in memory would put it at F800. This also turns out to be the easiest address for the hardware to accept as I will explain shortly.

The next problem is the software. However, North Star provides a MOVER program which can change the DOS for a different controller origin. I assume they also provide this for their other software such as CP/M and Pascal.

A trickier software problem is in the bootstrap PROM on the controller board. The bootstrap program can be reassembled and burned into another PROM but my relocation method will not require this. What will be required are some simple modifications to the DOS, some utilities, and to other programs which call the controller directly.

Modifying the Hardware

An examination of the schematic for the controller shows that the location the board occupies in memory is determined by the programming of the DSEL board select PROM. The standard PROM is programmed for address E8. (The DSEL PROM only responds to the high byte of the address) Address E8 means that address lines A11, A13, A14, and A15 are high and all other

lines, (of the high byte) including A12, are low.

Now what do you suppose the DSEL PROM would think if we inserted an invert gate between address line A12 and the PROM pin where A12 normally connects? You got it! Whenever A12 was low, the PROM would think it was high. And when A12 was high, the PROM would think it was low. A12 goes high for address F8. Voila! We have just relocated the controller to F800! There is even a spare inverter on the board for the purpose. It's IC 14B, pins 1 and 2. For those who can take care of the bootstrap problem, the modification is ended. Congratulations!

Now for the rest of us. We need that bootstrap program, at address E800, in order to get going. But once the DOS has loaded, the bootstrap is not used again. What we need is some way to switch the board to E800 for startup and then back to F800 for running bigger programs.

Oring Exclusively

The neat thing about IC logic is that, when some requirement pops up, you just reach up on the shelf and grab the one you need. All the circuit design has been done for you. What we require is a switchable inverter; one that will, at times, invert A12 and, at other times, pass A12 straight through. Such a device is called an exclusive OR gate. If we use an XOR gate instead of a simple inverter, in our relocation scheme, we can switch the controller address back and forth between E800 and F800 with some simple software commands. Wow! It's just like magic!

Uh--what's going to do the switching? I was afraid you were going to ask that! Well, the best choice would be an I/O port. If you are using a memory board that is equipped for bank switching, you can steal the I/O port from that board.

I'll use a North Star 32K memory board as an example.

The Software Switch

North Star's memory boards use an IC latch (flip-flop) to implement bank switching. On the 32K board, the latch is a 74LS74 at location 6C. Only one output of the latch is used, pin 9. The other output, pin 8, is not used. We can use pin 8 to switch our controller board.

The XOR gate we will use on the controller board will be a 74LS86. That will require the control signal to be a logic high for inversion to occur and a low for A12 to pass through non-inverted. Pin 8 of IC 6C can be made to toggle from a high to low by outputting the proper byte value from port C0.

Now, some of you are no doubt quick to realize that this will affect the memory board operation as well; but that is what we want. Consider this:

The whole purpose of the modification is to increase available memory. That means memory must normally be available up to address F7FF for use by applications programs. But what do we do with that memory when we switch the controller to E800 during startup? We must turn it off somehow, that's what. If we configure our memory board to be OFF during startup, pin 8 of IC 6C will be low and our controller will appear at address E800 in memory. The computer jumps to the boot PROM at E800 and off we go to another fine start.

Restrictions

As I have just described the modification there is one restriction. Only the top 8K or so of memory can be allowed to be disabled by the IC at 6C. This is so because there must be memory available to receive the first sector of the DOS. That may be low memory in the case of the DOS but other programs, such as CP/M, often load their DOS in high memory.

A better way to disable the memory, when switching the controller address, is to use the phantom signal. This has the advantage that up to an entire 64K can be made to disappear without any restrictions.

A complete description of phantom, and how to use it with the disk controller, is given in my article, "The Phantom Signal." (Compass Vol. 4, No. 1, p. 5) However, a brief summary of the necessary modifications is given in the following text.

Modifying the Software

The bootstrap program loads the first sector of the DOS. (512 bytes in DD) Thereafter, the DOS itself loads the remaining sectors. All we have to do is put a patch in the DOS to switch the controller back to F800 before loading the rest of the sectors. The DOS has been configured to think the controller is at F800 and through a little magic and trickery we have fooled the DOS into thinking the bootstrap program is also at F800.

The Connections and Patches

A 74LS86 XOR gate must be added to the controller board. The best way to do this is by piggybacking it onto another IC. A good candidate is the DSEL PROM itself, at location 11C, since the required pins line up quite well.

If you are nervous about soldering to the PROM (it is expensive!) the "dead bug" method can be used. This entails laying the 74LS86 on its back on top of the PROM with its "feet" pointing up. A small strip of double faced tape can be placed between them to hold things fast.

For jumpering, either lightly tack solder to the required pins or salvage some push on connectors from an old socket and use those. Soldering is best but using push on connectors will allow the board to be restored to its original condition. For the ground and +5V connections, use some circuit traces in the vicinity. See fig. 1. for more details.

Note: Following are the instructions to disable a 32K memory board by use of the phantom signal. If you can't or don't want to use phantom, see your memory board manual for the procedure to disable it by means of an I/O port.

1. Jumper pin 9 of IC 6C on the memory board to pin 2 of the 74LS86 on the controller board. The best way to do this is to position the boards in adjacent slots and jumper between them.

2. On the memory board, disable the bank switching by bending pin 5 of IC 6A out of the socket. Now, attach it to ground with a small jumper. (Pin 4 is connected to ground. The jumper can be attached there.)

3. Enable phantom by installing the "PE" jumper near IC 4E.

4. Enable phantom on the controller board by connecting a jumper from pin 6 of IC 7C to pin 67 on the S-100 bus strip. If pin 67 is missing, you can jumper directly to the "PE" pad on the memory board by putting it in an adjacent slot as in the step above.

5. Address the memory board to cover the area of E800 to F800. If you are not using any other memory mapped devices (monitor PROM, video board, etc.) you can address the memory for a full 64K.

6. The configuration header at location 7D should be connected as for bank switching. Any bit can be selected as the control bit but for ease of installation I will use bit 4. Disconnect the jumper from pin 3 to pin 1 and connect a jumper from pin 3 to pin 12. This will make the control byte 11H to switch the controller from E800 to F800 and 10H to switch it back. A RESET or power off/on will also switch the controller back to E800. For this to work, pin 6 should remain connected to pin 7 on the header.

Other Boards

North Star's 16K RAM boards are almost identical to the 32K boards and these modification steps should apply to those as well. If you are using all 16K RAM boards, only one need be modified, the topmost one. As for the HRAM, or other boards, I have no information but the following comments should help in configuring them.

The main requirement is to provide a low signal to pin 2 of the 74LS86 during startup. That puts the controller at E800. A high on pin 2 will switch it to F800. Any scheme that will do this under software control will work. The phantom signal from the controller board should work with most RAMs that recognize phantom.

The DOS Patch

All versions of North Star's DOS load in the same manner. The bootstrap program reads the first sector into RAM then jumps to the DOS startup address. The DOS itself loads the remaining sectors.

With DOS 2.1.1, the standard load address is 100H and the startup address is 10AH. Address 10AH contains a further jump to address 151H where the various registers are initialized with the proper values for loading the remaining sectors. The area from 151H to 181H is actually the stack area and the initial data stored there for startup will later be overwritten. We are going to put our patch there.

```
0100 =      DOS: EQU 100H
              ;
0151              ORG DOS+51H
              ;
0151 3E11      MVI A,11H
0153 D3C0      OUT 0C0H
0155 317701    LXI SP,177H
0158 AF        XRA A
0159 2C        INR L
015A C3F301    JMP 1F3H
```

The only real change is the loading of 11H into the A register and outputting it to port C0. That switches the controller to F800. The DOS should be configured, with the MOVER program, to expect the controller at F800 before the patch is entered.

Patching Other Programs

Other programs, such as CP/M, will have to be patched in like manner. There are so many different versions floating around that I will not attempt to give an example here. Remembering that the startup address will always be 0AH bytes in from the load address of the cold start

loader should enable you to find the proper patch area.

Some versions of CP/M, such as those from Lifeboat Associates, have an auto relocating algorithm which patches the BIOS for the proper controller address during bootup. Such versions will have to have the algorithm adjusted or it will erroneously decide that the controller's address is, and will remain at, E800.

(The algorithm takes the address which is left in the HL register pair, after jumping from the bootstrap PROM to address XX0AH of the CP/M cold startloader, and computes the controller address from it. Look for the code sequence: EB (XCHG), 22 XX XX (SHLD) somewhere after the entry point at address

XX0AH.)

A few programs, such as North Star's Monitor, have a command to bootload the disk. These programs must be patched to switch the controller back to E800 before booting or the system will crash. That only requires that a 10H be output from port C0 before jumping to E800.

I encourage users who try this relocation method, and find the required patches in various programs, to share their knowledge with other INSUA members through Compass.

For any application where you want the controller to remain at E800, just use the standard unmodified DOS or other software.

Happy computing!

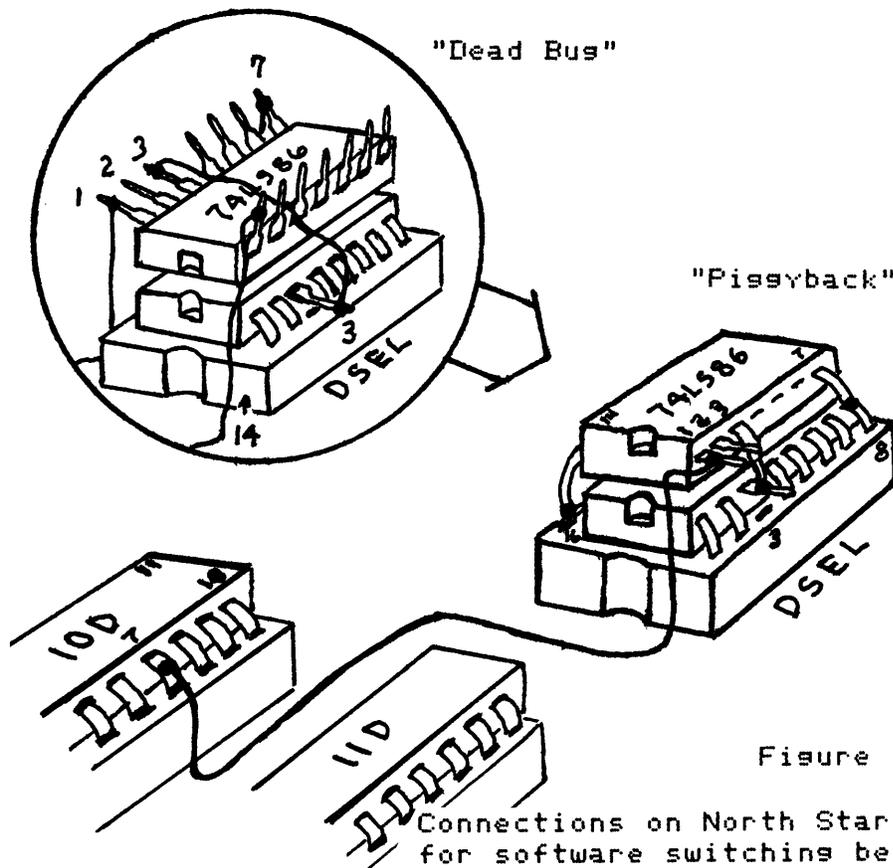


Figure 1.

Connections on North Star DD controller board for software switching between E800 and F800.

1. Jumper pin 7 IC 10D to pin 1 of 74LS86. (A12 in)
 2. Bend pin 3 of DSEL out of the socket and connect to pin 3 of the 74LS86. (A12 out)
 3. Solder pin 7 of the 74LS86 to pin 8 of DSEL or connect to a nearby ground. (ground)
 4. Solder pin 14 of the 74LS86 to pin 16 of DSEL or connect to a nearby +5V. (+5V)
 5. Connect the signal from the memory board to pin 2 of the 74LS86. (control)
- No other pins of 74LS86 are used.

A File Conversion Program for North Star

305-
757-8110

Doug Ingram
530 N. E. 91 Street
Miami Shores, FL 33138

Send diskette
with copy of program
to Doug Ingram

If you are using North Star DOS and CP/M 2.2, sooner or later you will want to transfer files from DOS to CP/M. NSCVRT is a handy program for this purpose. The North Star double density controller has been around about five years. Certainly one of the reasons for its popularity has been the fine BASIC which North Star has provided with its computers. Over the years I have acquired a large library of DOS BASIC files. As my system grew and CP/M became my preferred operating system it was necessary to convert my DOS files to CP/M. NSCVRT was written for this purpose.

SYSTEM REQUIREMENTS

NSCVRT requires a North Star double density controller and at least two drives. The two popular versions of CPM 2.2 for the North Star are supplied by Lifeboat Associates and North Star Computers. Either may be used with NSCVRT. Near the beginning of the program there is an 'LBOAT EQU' which must be set true if you are using this program with Lifeboat Associates CP/M. Set this EQU false if used with North Star CP/M. This program will assemble with the CP/M-supplied assembler 'ASM'. After assembly a 'COM' file may be created with the LOAD program. Upon execution a prompt will appear for read and write drives. The user will be prompted for each file to be transferred.

Single density files will cause a BIOS error. To convert single density files, the North Star program "CO" (compact disk) should be run and the convert to double density option should be selected. NSCVRT will convert all DOS files to CP/M, but only BASIC and BASIC data files will operate with CP/M. DOS type 0 and 1 are machine language programs which will not interface with CP/M's BDOS and will not operate successfully with CP/M.

HOW NSCVRT WORKS

The DOS directory is recorded on the first eight 256-byte blocks (2048 bytes) of the disk. Each entry is 16 bytes long. This allows 128 directory entries per disk. The figure below describes a DOS entry.

NORTH STAR DOS DIRECTORY

	NAME....	DA	BL	T	LA.
	^	^	^	^	^
	:	:	:	:	:
	:	:	:	:	:
8 byte file name	-----+	:	:	:	:
2 byte beginning disk address	-----+	:	:	:	:
2 byte count of 256 byte blocks	-----+	:	:	:	:
1 byte file type	-----+	:	:	:	:
3 byte load address / size of BASIC program	-----+	:	:	:	:

NSCVRT loads the DOS directory to memory, after which a search takes place for active entries. An active directory entry will contain a blank space (20h) character as the first byte of the directory entry. Bytes DA (see figure above) are converted to a

CP/M format of track/sector. A call to BIOS is then made to read a 128-byte CP/M record. This procedure is repeated until the entire file has been read. Bytes BL indicate the file size and are used to determine how many 128-byte CP/M records are to be read. The T type byte is used to determine the CP/M extension to be added to the file. NSCVRT uses BIOS calls to read the DOS disk and BDOS functions to write the file in a CP/M format. BIOS calls are necessary because of the DOS directory address and the need to access the first two tracks of the disk. CP/M reserves the first two tracks as system tracks. CP/M's BDOS will not access these tracks. Due to differences in implementation of the BIOS, it is necessary to set the beginning sector to 0 for North Star CP/M. It is also necessary to cancel the effect of disk skew. North Star BIOS is written to use a disk skew of 5 with 512 byte blocks. Since DOS uses no skew it is necessary to cancel the effect of skew. The DOSXLT table is used as a look up table to translate records so they may be used by the BIOS to read the DOS disk in sequential order. This table is not used for Lifeboat CP/M.

NSCVRT should eventually be available as an INSUA disk, but until its announcement it is available from the author for a charge of \$15.00. Please specify 5-1/4" North Star double density or 8" single density. (The source assembly code printed below has been compressed by the elimination of most multiple carriage returns in order to save space--Ed.)

```
;NSCVRT.ASM
;JUNE 17, 1983
;Corrections complete for North Star CP/M
;   changed sect to start at 0 in direct, addr, ldbuf
;   skew is required for North Star CP/M
;JUNE 18, 1983
;Added routines to prompt for file transfer
;JUNE 19, 1983
;Added routines for the 'if' assembly of Lifeboat and North Star
;   CP/M this will allow either CP/M to be used with this
;   program.
;   This program will allow files to be transferred from the North
;   Star DOS operating system to CP/M. After transferring a basic
;   interpreter such as BAZIC (available from Micro Mike) may be used
;   to run DOS files under the CP/M operating system. While all DOS
;   files may be transferred only basic and basic data files have
;   practical application for use with CP/M.
;   To use this program an editor should be used to change the 'lboat'
;   equate to true or false. The 'lboat' equate should be set true
;   if you are using the LIFEBOAT ASSOCIATES version of CP/M. If you
;   are using the North Star version of CP/M as supplied by NORTH STAR
;   COMPUTERS 'lboat' should be set false. Then this file may be
;   assembled with the CP/M supplied assembler 'ASM'.

;   This program was written for use with the North Star double density
;   disk controller and CP/M 2.2. A single density DOS file will cause
;   an error. If you wish to transfer single density files first use
;   the DOS program CO to compact the disk and select the double density
;   option.
;   Please forward all comments, suggestions, and improvements to:
;   Doug Ingram 530 N. E. 91 Street Miami Shores, Fl 33138
true   equ   0ffffh
false  equ   not true
;
;   >>> SET THE FOLLOWING EQU TO YOUR SYSTEM NEEDS <<<
;   *****
lboat  equ   false           ;set lboat TRUE if using Lifeboat CP/M
;                                     ;FALSE if using North Star CP/M

        if    lboat
nscpm  equ   0
lbt    equ   1
        endif
        if    not lboat
nscpm  equ   1
lbt    equ   0
        endif
org    100h           ;beginning of tpa

wstart equ   0           ;warm start
bios    equ   0001h     ;BIOS warm start address
consinn equ   06h-3     ;BIOS console input function
seldskn equ   1bh-3     ;BIOS select disk function
settrkn equ   1eh-3     ;BIOS set track function
setsecl equ   21h-3     ;BIOS set sector function
setdman equ   24h-3     ;BIOS set dma address function
```



```

call      conin
ani      05fh
cpi     04lh
jc      input1
cpi     046h
jnc     input1
sui     'A'
cpi     04
jnz     input2
lxi     d,exit
jmp     restart
input1   pop      d
input2  pop      d
ret

;This routine will read the North Star directory to memory
direct  lxi     h,dirbuf
shld   buf
mvi   a,0
sta   trk
lda   rdrive
mov   e,a
call  select
lda   rdrive
mov   c,a
call  seldisk
mvi  a,1bt
sta  sect
call  readsec
lhld  buf
lxi  d,128
dad  d
shld  buf
lda  sect
inr  a
sta  sect
cpi  16*1bt
jc  rdsec
ret

;The following routine will move the hl register pair to the
;required byte of the 16 byte file entry. The hl will point
;to byte 0 of the current entry and will be incremented by
;the value passed to this routine in the c register.
dirbyte  lhld  dirblk
mvi  b,0
dad  b
ret

;The following routine will advance to the next file entry
;and store the current value at dirblk. A test is made to
;determine if all directory entries have been used. A value
;of 0ffh is returned in the accumulator to indicate directory
;search complete.
nextdir  lhld  dirblk
lxi  b,16
dad  b

```

```

;get character typed
;convert lower to upper case
;is character < A
;if so do again
;is character > E
;if so do again
;subtract 4lh from value
;if it was E it will now be 4
;not E return with value in acc
;de points to message
;do system restart

shld  dirblk
lda  entryct
cpi  128
rc
lxi  d,normal
jmp  restart
;The following routine will prompt for input and will allow files to
;be selected for transfer
dpfile lxi  d,crlf
lxi  h,fcbn
xra  a
mov  c,a
mov  e,m
dpfile1 mov  h,b
push b
call  conout
pop  h
pop  h
inr  h
inr  c
mov  a,c
cpi  8
jnz  dpfile1
lxi  d,prompt
call  print
conio 0
cpi  0
jz   dpfile2
ani  05fh
cpi  'C'
jz   dpfile3
cpi  'S'
jz   dpfile4
cpi  'E'
jz   dpfile5
jmp  dpfile2
e,ra
conout
call  conout
xra  a
ret
dpfile4 mov  e,a
call  conout
mvi  a,0ffh
ret
dpfile5 mov  e,a
call  conout
jmp  cpm
;the following routine will place a name in the file control block
;in preparation for a write operation
fname  lda  entryct
inr  a
sta  entryct
lhld  dirblk
mvi  c,0

```

```

;store new value
;load count of directory entries
;directory holds 128
;return if more
;point de to conclude message
;out message and re-boot
;print one blank
;line
;point to file name to print
;zero
;counter
;get a byte

;print to console

;next byte
;bump counter

;have we printed 8 yet ?
;if no loop
;display input prompt

;get character
;anything typed ?
;if not loop
;change to upper case
;copy ?

;skip ?
;exit ?

;not legal, loop
;prepare to print
;print it
;zero the next flag

;set the next flag

;return to cpm
;the following routine will place a name in the file control block
;in preparation for a write operation
;current dir entry
;bump counter
;store new value
;point hl to file name
;set counter

```

```

name1
    lxi    d,fcbn
    mov    a,m
    cpi    20h
    jz     next1
    stax   d
    inr    c
    mov    a,c
    cpi    8
    rz     next2
    inx    d
    inx    h
    mov    a,m
    ;cpi 20h
    ;next2
    ;cpi '?'
    ;next3
    ;cpi '*'
    ;next3
    ;cpi ':'
    ;next3
    ;cpi '.'
    ;next3
    ;cpi ' '
    ;next3
    jmp    name1
    next1 mvi    a,0f1h
    next2 xra
    next3 mvi    a,'x'
    ret
;add extensions to CP/M file control block
;type 0 or 1 will be com extensions
;type 2 is 002 extension
;type 3 is 003 extension
;and so on through 9
;all others are undefined (UDF)
type    mov    a,m
ani     127
cpi     0
jz      com1
cpi     1
jz      com1
cpi     2
jz      bas1
cpi     3
jz      dat1
cpi     4
jz      ty4
cpi     5
jz      ty5
cpi     6
jz      ty6
cpi     7
jz      ty7
cpi     8
jz      ty8
;point to file control block
;load acc with character
;is first character a space ?
;if so go to next directory entry
;store character at fcb
;bump counter
;8 characters for a file name
;return if we have all 8
;increment to next
;load next character
;check if legal CP/M file character
;question mark not allowed
;asterisk not allowed
;colon not allowed
;period not allowed
;loop to store and continue
;not a current directory entry
;replace space with null
;replace ?,*,.,. with X
;load acc with file type
;mask high order bit
;/N/S operating system file
;if true label this file com
;/N/S assembly file, go file
;label it com for CP/M
;/N/S basic program
;label it .002 for CP/M
;/N/S data file
;label it .003 for CP/M
;point to extension position
;zero counter
;load acc with first character
;store acc
;bump counter
;add one to count
;count to acc
;have we reached 3 ?
;if no loop
;return to calling routine
;The following routine uses the N/S directory to determine disk
;address to read. Bytes 8 & 9 of the N/S directory contain this
;information. The N/S operating system numbers sectors consecutively
;0-349 for single sided double density. This is the scheme N/S
;uses to allocate files. CP/M on the other hand numbers track then
;records/sectors. A CP/M disk would start with track 0 sector 1
;this count would continue through track 34 sector 10 which would be
;the inner most track on the disk.
;One N/S track contains 20 256 byte blocks in order to convert to
;CP/M this must be changed to 40 128 byte records per track.
;The following routine converts N/S file address to a CP/M format
;of track-records and sets up counters to convert from a 256 byte
;structure to a 128 byte record as required by CP/M.
    cpi    9
    jz     h,udfex
    lxi    h,comex
    jmp    h,comex
    lxi    h,basex
    jmp    h,basex
    lxi    h,datex
    jmp    h,datex
    lxi    h,ty004
    jmp    h,ty004
    lxi    h,ty005
    jmp    h,ty005
    lxi    h,ty006
    jmp    h,ty006
    lxi    h,ty007
    jmp    h,ty007
    lxi    h,ty008
    jmp    h,ty008
    lxi    h,ty009
    jmp    h,ty009
    exten d,fcbe
    mvi    b,0
    mov    a,m
    stax   h
    inx    h
    inr    d
    mov    a,b
    cpi    3
    jnz   exten1
    ret
;The following routine uses the N/S directory to determine disk
;address to read. Bytes 8 & 9 of the N/S directory contain this
;information. The N/S operating system numbers sectors consecutively
;0-349 for single sided double density. This is the scheme N/S
;uses to allocate files. CP/M on the other hand numbers track then
;records/sectors. A CP/M disk would start with track 0 sector 1
;this count would continue through track 34 sector 10 which would be
;the inner most track on the disk.
;One N/S track contains 20 256 byte blocks in order to convert to
;CP/M this must be changed to 40 128 byte records per track.
;The following routine converts N/S file address to a CP/M format
;of track-records and sets up counters to convert from a 256 byte
;structure to a 128 byte record as required by CP/M.
    addr  mov    c,m
    inx   h
    mov   b,m
    push h
    xra  a
    cma
    mov  d,a
    mvi  a,0ah
    cma
;point to default extension
;routine to put extension on file

```

```

mov e,a
inx d
lxi h,0
mvi a,17
push h
dad d
inc d
xthl
pop h
push psw
mov a,c
ral
mov c,a
mov a,b
ral
mov b,a
mov a,1
ral
mov l,a
mov a,h
ral
mov h,a
mov psw
dad a
jnz a
ora a
mov a,h
ral
mov d,a
mov a,1
ral
adl lbt
sta sect
lxi h,trk
mov m,c
pop h
;get total # of N/S blocks X 2 and store at blocks to read
inx h
mov c,m
inx h
mov b,m
push h
lxi h,0
dad b
dad h
dad h
shld btoread
pop h
ret
;The following routine will read the North Star disk and set up
; a buffer to hold up to one track of data.
ldbuf lxi h,fbuf
shld buf
mvi a,0
sta bnfct

```

```

;drive to read N/S disk
;tell bios N/S drive

rddrive
c,a
selqsk
bufct
a
40
bufct
btoread
h
btoread
a,h
Offh
zeroct
readsec
buf
d,128
d
;add one record to hl
;store new value
;get sector count
; bump counter
;store new value back to memory
;40 records/track
;loop

;load track value
; bump counter
;store new value
;start with first sector
;store new value

;zero count

;The following routine preforms the read operation. A 128 byte
;CP/M record will be read by the readsec routine.
readsec ldx c,a
mov b,0
call settrk
ldx sect
if not lboat
call deskew
endif
mov c,a
mvi b,0
call setsec
lxi h,buf
mov c,m
inx h

```

```

mov b,m
call setdma
jmp read
;North Star CP/M uses disk skew, however North Star DOS does not.
;In order to read DOS files a translate must be used to cancel
;disk skew. The routine below uses the 'dosxit' table to de-skew
;DOS files.
deskew lxi h,dosxit
mov d,0
mov e,a
dad d
mov a,m
ret
;The following will open a file on the CP/M drive.
openfil lda wrdrive
mov c,a
call seldsk
xra a
sta fcbcr
sta extent
sta s2
lda wrdrive
inr a
sta fcb2
lxi d,fcb2
call delete
lxi d,fcb2
call make
lxi d,nodir
cpi 255
inr a
call restart
;The following will copy from the file buffer to the CP/M drive.
copy lda wrdrive
mov c,a
call seldsk
lxi h,buf
lxi d,fbuf
mov m,e
inr h
mov m,d
lda bufct
cpi 0
jz sdma
call d,fcb2
call write
ora a
lxi d,space
cnz restart
lhld buf
lxi d,l28
dad d
shld buf
xchg
mov b,m
call setdma
jmp read
;function call, buffer addr
;North Star CP/M uses disk skew, however North Star DOS does not.
;In order to read DOS files a translate must be used to cancel
;disk skew. The routine below uses the 'dosxit' table to de-skew
;DOS files.
deskew lxi h,dosxit
mov d,0
mov e,a
dad d
mov a,m
ret
;The following will open a file on the CP/M drive.
openfil lda wrdrive
mov c,a
call seldsk
xra a
sta fcbcr
sta extent
sta s2
lda wrdrive
inr a
sta fcb2
lxi d,fcb2
call delete
lxi d,fcb2
call make
lxi d,nodir
cpi 255
inr a
call restart
;The following will copy from the file buffer to the CP/M drive.
copy lda wrdrive
mov c,a
call seldsk
lxi h,buf
lxi d,fbuf
mov m,e
inr h
mov m,d
lda bufct
cpi 0
jz sdma
call d,fcb2
call write
ora a
lxi d,space
cnz restart
lhld buf
lxi d,l28
dad d
shld buf
xchg

```

10

```

lda bufct
dcr a
sta bufct
jmp copy1
;Close file routine for the CP/M drive.
closfil lda wrdrive
mov c,a
call seldsk
lxi d,fcb2
call close
lxi h,wrprot
inr a
call restart
ret
delete mvi c,deletef
jmp bdos
select mvi c,selectf
jmp bdos
make mvi c,makef;
jmp bdos
sdma mvi c,sdmaf
jmp bdos
write mvi c,wrtf
jmp bdos
esdsk mvi c,ressdsk
jmp bdos
close mvi c,closef
jmp bdos
conin mvi c,coninf
jmp bdos
conout mvi c,conoutf
jmp bdos
print mvi c,prntf
jmp bdos
default mvi c,defltf
jmp bdos
conio mvi e,0ffh
jmp bdos
comex db 'COM'
basex db '002'
datex db '003'
ty004 db '004'
ty005 db '005'
ty006 db '006'
ty007 db '007'
ty008 db '008'
ty009 db '009'
udfex db 'UDF'
sect ds 1
trk ds 1
btoread dw 0000h
buf dw 0000h
bufct ds 1
;count of records in buffer
;store new value
;de points to fcb
;bdos close funtion
;write protected disk
;255 becomes 0
;if error restart
;function # to delete a file
;bdos will supply return to calling
;extensions used to
;indicate file
;type
;sector count
;track count
;count of CP/M sectors to read this file
;buffer address
;count of records in buffer

```

11

```

dirblk ds      2      ;pointer to memory location of dir entry
entryct db      0      ;number of file entry (128 max)
rddrive ds      1      ;read drive
wrddrive ds      1      ;write drive
deflts ds      1      ;storage location for default disk
                rdmsg
                wrmsg
                hold
                prompt
                exit
                samedr
                nodir
                space
                wrprot
                normal
                dirbuf ds      2048      ;directory buffer
                rdbuf ds      5120      ;buffer for file
                ds      64      ;32 level stack
                end
;file control block follows
fcb2 ds      1      ;disk drive to write
fcbn ds      8      ;name of CP/M file
fcbf ds      3      ;extension
extent ds      1      ;CP/M extent number
s1 ds      1      ;s1 byte
s2 ds      1      ;s2 byte
rext ds      1      ;record count
bx ds      16      ;blocks in this file
fcbcr ds      1      ;current record
ds      3      ;random access bytes
dphaddr dw      0
xitaddr dw      0
;The following table is used to de-skew a DOS disk, only used with
;North Star CP/M.
dosxlt db      0,1,2,3
        db      8,9,10,11
        db      16,17,18,19
        db      24,25,26,27
        db      32,33,34,35
        db      4,5,6,7
        db      12,13,14,15
        db      20,21,22,23
        db      28,29,30,31
        db      36,37,38,39
        db      cr,lf
        db      cr,lf,'$'
        db      cr,lf,cr,lf
        db      'NSCVRAT Version 1.0 ',cr,lf
        if      lboat
        db      'Assembled for use with Lifeboat Associates CP/M 2.2'
        endif
        if      not lboat
        db      'Assembled for use with North Star Computers CP/M 2.2'
        endif
        db      cr,lf,lf

```

```

'For 35 track CP/M on North Star double density',cr,lf
'disk. Files may be transferred from a double',cr,lf
'density North Star format to a CP/M format disk.',cr,lf
cr,lf,cr,lf,'$'
;
cr,lf,'Enter drive to read N/S format disk',cr,lf
        (A,B,C,D) or (E)xit > '$'
;
cr,lf,'Enter drive to write CP/M format files',cr,lf
        (A,B,C,D) or (E)xit > '$'
;
cr,lf,cr,lf,'Strike any key when ready > '$'
cr,lf,'Enter Copy, Skip, or Exit',cr,lf,'$'
cr,lf,'Returning to CP/M',cr,lf,'$'
cr,lf,'Cannot use same drive to read and write files',cr,lf,'$'
cr,lf,'No directory space available',cr,lf,'$'
cr,lf,'Out of data space for this disk',cr,lf,'$'
cr,lf,'Disk is write protected',cr,lf,'$'
cr,lf,'Transfer complete',cr,lf,'$'
;
                2048      ;directory buffer
                5120      ;buffer for file
                64      ;32 level stack
                start

```

A North* BASIC Pre-Processor

By Jim Shearer

(The author published an article in May 1984 Microsystems on a preprocessor for MBASIC. Here is his announcement of his copyrighted preprocessor for North Star BASIC. The purchase price of NSPRE is \$15.00. Write Mr. Shearer c/o Tri-State Digital, 210 West Maumee Suite 2, Angola, Indiana 46703 (219) 665-7673 --Ed.)

1. SUMMARY

NSPRE is a pre-processor for CP/M based North Star BASIC and North Star look-alikes such as Micro Mike's baZic. It allows BASIC programs to be written without line numbers and with long variable names on a text editor or word processor. The resulting ASCII text is converted by NSPRE into a compressed run-time program with remarks and white space removed, line numbers added, and long variable names replaced by single letters or a letter followed by a single digit.

NSPRE is written in C and was compiled using C/80, a copyrighted C compiler from The Software Toolworks.

2. USAGE

NSPRE is a CP/M .COM file and is run by typing NSPRE followed by the name of the source file and, optionally, the output file name and the listing device. For example,

```
nspre amortize.baz amortize.002 lst:
```

will take the ASCII input file "amortize.baz" and convert it to a North Star run-time file "amortize.002" listing the variable assignments and label table on the printer.

The general form of the NSPRE command is:

```
nspre srcfile.ext outfile.ext device
```

where "outfile.ext" and "device" are optional. If "outfile.ext" is not specified, the run-time program will be written to a file named "srcfile.002". "Device" will normally be either "con:" for console

listing or "lst:" for printer listing. If "device" is not specified, the listing will go to the console.

The file types (".ext") can be anything you choose. The type of "outfile" will normally be the type required by your BASIC interpreter. For example, baZic requires program files of type .002. **The srcfile and outfile must have different names or different types.**

The variable assignment listing and label table can be directed to a disk file by specifying a file name for "device". The listings will be written to the file in formatted ASCII.

Note that any or all parts of the command line can be typed in lower case or upper case. Except for data enclosed in quotes and print format strings, NSPRE converts all lower case letters to upper case.

3. VARIABLE NAMES

Variable names consist of letters, digits and the underline. The first character must be a letter. The last character must be '\$' if the name represents a string variable. Names can be of any length, however NSPRE distinguishes only the first eight characters. For example, the following variable names would refer to the same variable as far as NSPRE is concerned:

```
LAST_NAME$
```

```
last_name2$
```

(Remember, all lower case letters are converted to upper case.)

A variable name may not be the same as any of the BASIC reserved words. However, reserved words can be embedded in a variable name. For example, the

name "PRINT" is not allowed, but "PRINT2" is. **A variable name must never begin with the letters FN since those letters are reserved for functions.**

Names for numeric variables and those for string variables may be the same except for the trailing \$. For example, DATE and DATE\$ would be distinguished by NSPRE as representing two different variables. However, array names and function names must not be the same as other names. For example, you must not use ITEM and ITEM() in the same program. Similarly, DATE\$ and FNDATE\$ must not be used together.

4. LABELS

Labels are used in branching statements in place of line numbers. **Line numbers may not be used.** Label names follow the same rules as variable names except that the first character must be the at-sign, '@'. Examples:

```
GOOD  
@EOJ  
@print_mod
```

```
BAD  
END_JOB (@ missing)  
@print/sr (/ is illegal)
```

Note that only the first eight characters (@print_m) of the label "@print_mod" would be recognized.

The line to which control is transferred by the GOTO, GOSUB, etc. statements, must have the label name as the leftmost entry on the line. For example, suppose you have a branching statement of the form:

```
GOTO @CALC_MOD
```

At some other point in the program, you must have a line beginning with the label @CALC_MOD.

BASIC statements that require labels include GOTO, GOSUB, RESTORE, EXIT, ERRSET (optional label) and IF ... THEN (optional label).

5. REMARKS

Since NSPRE eliminates all remarks and unnecessary "white space", they should be used freely in your source program. They will not penalize the run-time performance and they will make program maintenance much easier. Remarks may begin with the usual REM statement or the semicolon (;). When either of those is encountered by NSPRE, it and all else that follows on the same line is omitted from the run-time file. Also, spaces, blank lines and tabs are all eliminated. So you can use indentation, block separation, etc. freely without slowing down or increasing the size of the run-time program.

6. HINTS

NSPRE assigns variable names to the letters A...Z, A0...Z0, ..., A9...Z9, as they are encountered. For example, if your program has 27 distinct numeric variable names, the first 26 would be assigned the letters A through Z and the 27th would be assigned A0. Since BASIC handles the single letters faster than the letter-digit combinations, high-use variables should be named early in the program. For example, if your program uses the name INDEX frequently, you should have a line early in the program of the form:

```
INDEX = 0
```

Although the initialization may be unnecessary in your program, such a statement will insure that INDEX is assigned one of the single letter variables.

7. EXAMPLE

Listings 1 and 2 show the source and run-time files for a simple Amortization program. Note that in Listing 1, lower case letters were used in the program lines. NSPRE converts those to upper case as shown in Listing 2.

The variable assignment tables and label table shown at the top of Listing 2 are produced by NSPRE. The rest of the listing was not. It was listed using the BASIC LIST command. Note in Listing 2

that lower case letters inside quotes are not converted to uppercase.

8. LIMITATIONS

Micro Mike's baZic has a useful APPEND statement that can be used at run

time. Unfortunately, the present version of NSPRE does not provide for that feature. You must include in your source file any subroutines or functions that you might otherwise have in a separate program to be appended at run time.

###

```
; <amortize.baz> [Some double-spacing eliminated--Ed.]
; Prints amortization table
cls$ = chr$(27) + "*";           # televideo screen clear #
prt_port = 1;                     # printer port #
    print cls$
    print\print

    input"Amount of loan: ",amount
@int_inp  input"Interest rate (%): ",i
    if i<1 then @int_inp;         # check that % entered #
    i = i/1200;                   # monthly rate #
    input"Period in months(0 if not known): ",period
    if period = 0 then @pay_inp
    payment = int(100*i*amount/(1 - 1/(1 + i)^period) +.5)/100
    goto @main
@pay_inp  input" then enter payment: ",payment
    period = int(log(1-amount*i/payment)/log(1/(1+i))+.5)
@main     input"Output device (S)creen, (P)rinter): ",dev$
    if dev$="P" then dev=prt_port else dev=0
    if dev$="p" then dev=prt_port

    gosub @header
    balance = amount
    for month = 1 to period + 1
        interest = int(i*balance*100+.5)/100
        principal = payment - interest
        if principal > balance then principal = balance
        balance = balance - principal
        print#dev, %10I,month,%16F2,principal+interest,principal,
        print#dev,%16F2,interest,balance
        if balance = 0 then exit @eoj
        linecnt=linecnt+1
        if month/6<>int(month/6) then @main_nxt
        print#dev\linecnt=linecnt+1
        if linecnt>55 then gosub @page_adv
@main_nxt next
    goto @eoj
page_adv  for j = 1 to 66 - linecnt
    print#dev
next
@header  print#dev,tab(30),"AMORTIZATION SCHEDULE"\print#dev
    print#dev, %C$18F2,AMOUNT," LOAN AT",%#,1200*i,
    print#dev,"% INTEREST",%C$10F2,PAYMENT," MONTHLY PAYMENT"
    print#dev
    print#dev,"          MONTH          PAYMENT          PRINCIPAL          "
    print#dev,"          INTEREST          BALANCE"
    linecnt=5
    return
@eoj     end
```

Listing 1. amortize.baz, source file for NSPRE

NUMERIC VARIABLE ASSIGNMENTS

A=PRT_PORT B=AMOUNT C=I D=PERIOD E=PAYMENT F=DEV
 G=BALANCE H=MONTH I=INTEREST J=PRINCIPA K=LINECNT L=J

STRING VARIABLE ASSIGNMENTS

A\$=CLS\$ B\$=DEV\$

LABEL TABLE

60 @INT_INP 130 @PAY_INP 150 @MAIN 320 @MAIN_NX 340 @PAGE_AD
 370 @HEADER 450 @EOJ

```

10A$=CHR$(27)+"*"
20A=1
30PRINTA$
40PRINT\PRINT
50INPUT"Amount of loan: ",B
60INPUT"Interest rate (%): ",C
70IFC<1THEN60
80C=C/1200
90INPUT"Period in months(0 if not known): ",D
100IFD=0THEN130
110E=INT(100*C*B/(1-1/(1+C)^D)+.5)/100
120GOTO150
130INPUT" then enter payment: ",E
140D=INT(LOG(1-B*C/E)/LOG(1/(1+C))+.5)
150INPUT"Output device (S)creen, P)rinter): ",B$
160IFB$="P"THENF=AELSEF=0
170IFB$="p"THENF=A
180GOSUB370
190G=B
200FORH=1TOD+1
210I=INT(C*G*100+.5)/100
220J=E-I
230IFJ>GTHENJ=G
240G=G-J
250PRINT#F,%10I,H,%16F2,J+I,J,
260PRINT#F,%16F2,I,G
270IFG=0THENEXIT450
280K=K+1
290IFH/6<>INT(H/6)THEN320
300PRINT#F\K=K+1
310IFK>55THENGOSUB340
320NEXT
330GOTO450
340FORL=1TOD66-K
350PRINT#F
360NEXT
370PRINT#F,TAB(30),"AMORTIZATION SCHEDULE"\PRINT#F
380PRINT#F,%C$16F2,B," LOAN AT",%#,1200*C,
390PRINT#F,"% INTEREST",%C$10F2,E," MONTHLY PAYMENT"
400PRINT#F
410PRINT#F,"            MONTH            PAYMENT            PRINCIPAL            ",
420PRINT#F,"            INTEREST            BALANCE"
430K=5
440RETURN
450END
    
```

NSPRE: A Review

By Saul G. Levy

(We asked Saul Levy to review NSPRE for Compass: here is the result --Ed.)

Why Structured BASIC?

Pascal and Forth programmers are always saying that the BASIC language is unstructured, and therefore, hard to use. They are correct in the sense that using short, obscure (to some!) variable names and GOTOs and GOSUBs make programming even harder than it has to be. I find BASIC trivial to use and after much practice have been wishing for longer variable names and word labels instead of the line numbers (I do prefer line numbers for editing). Adding some structure to a large or complicated program is very helpful. This is a review of a program which allows you to use an editor to write this type of structured BASIC program, then convert the source code into something that a North Star-compatible BASIC can execute.

A Useful Example

Figure 1 shows a simple program I wrote in 1980 to print out the individual bytes in a data file. It not only does not look like BASIC, it has no line numbers! Of course, this is not BASIC at all! It is a pseudo North Star BASIC.

The program name looks like F-BREAD, but it really stands for F-B-READ (file byte read). I call it F-BREAD, of course! This program is very useful to any programmer who has slaved over trying to figure out what s/he did wrong while writing a random-access data file program. FBREAD will read any file type.

Line numbers are replaced by labels which start with the '@' symbol in column 1. The labels are needed only when you want to interrupt the flow of the program and jump (GOTO or GOSUB) to the labelled line of code. Label names (and variables) may be typed in upper or lower case (note the reference to '@MAIN' four lines before the label '@main'). References to each label can appear in any location in the source code.

Long label names should make it clear what each routine is for. The labels and variables can be of any length, but only the first eight characters are important to the pre-processor (I explain why later on). For example: 'longlabel' and 'longlabeltwo' are both shortened to 'longlabe' by the pre-processor and refer to the same variable. Using such long names will cause confusion (due to the forming of duplicates) and are harder to keep track of (besides causing problems with keeping the columns of code lined up). I used abbreviations to keep the size of labels and variables down. Look at OUTDEV which stands for OUTPUT DEVICE. Using OUTPUTDEVICE or OUTPUT-DEVICE is ugly and uses up a lot of room on a line. (The hyphen '-' is illegal. It will represent the rubout key's underline character, 95 decimal = 5F hex, which my word processor cannot print.) You can make your own decision on which style to use. I could also have used FNO, FTYPE, FNAME\$, and FSIZE for FILENO, FILETYPE, FILENAME\$, and FILESIZE.

Variables must start with a letter, and if a string, must end with a '\$'. BYTE and BYTE\$ are two separate variables in the same way that North Star BASIC would use B and B\$. You can also embed numbers and the underline character (LOOP2 or LOOP-DE-LOOP, for example). North Star BASIC allows you to use BYTE and BYTE() as two different variables, but the pre-processor will not allow this. (The first is a simple variable, the second a DIMensioned variable.) byte and BYTE will refer to the same variable because all lower case letters outside of a pair of quote marks (") will be upper cased during pre-processing.

I used all upper case variable names because I find them easier to keep track of (except in the '@start' routine where they are of mixed case as an example). I used lower case for the BASIC key words (rem, dim, input, etc.) so as not to confuse them with my variable names.

You cannot use INPUT or input as a variable since it is a key word which is reserved by BASIC. You may use INPUT2 or input2 though. All variables starting with FN are reserved for user-defined functions and the variable name used with FN cannot be used as a regular variable (unlike North Star BASIC). This means you can use FNINPUT2, but cannot also use INPUT2 in the same program. These are minor limitations of the pre-processor that you must follow.

Note the use of a ';' for rem after FILENO=0. The pre-processor accepts either format, but the backslash after OUTDEV=0 will cause a problem and must be left out of the source code as we will see in the executable program (shown in Figure 3).

All REMs and unnecessary spaces will be removed from the source code by the pre-processor (the source code is crunched). Blank lines and tabs will also be removed so you can freely format the source code in any way that you want to (some of the tabs do cause a problem, however).

Time to Rewire Your Brain!

The rest of Figure 1 follows North Star BASIC usage. I found it took a considerable rethinking to convert my original program into the source code shown. I hope that practice will make this job easier. I believe that large programs will profit by the use of long labels and variable names. I also believe that you have to work harder to understand the source code because it is not BASIC code. My directory catalog program (DIRCAT) will give me a good test of these assumptions. The variable names have gotten totally out of control due to a mistake I made (see below) and I plan to rewrite DIRCAT in source code for use with this pre-processor.

The Pre-processor

The pre-processor program is called NSPRE, a 21K .COM file for use with CP/M. The C source code (not included) must be over 100K in size (usually 5-6 times larger than the assembled code).

This is almost as big as the Small C compiler is!

The package also comes with printed documentation, a .DOC file of the same information, and the amortization program as the example. The amortization program includes the source code file AMORTIZE.BAZ and the pre-processed file AMORTIZE.002 which is for use with Micro Mike's baZic (the latter's file format is exactly the same as a North Star BASIC program file and can be moved over to a North Star DOS diskette and run with North Star BASIC).

The upper limit of eight characters in the label and variable names is due to NSPRE's being written in Small C. NSPRE's C source code was compiled by C/80 into assembly source code which is then assembled and loaded. When the C source code is compiled with C/80, each label becomes an assembly routine label which many assemblers will limit to a maximum of eight characters. This limitation also applies to any (in this case, variable) names you run with a C program such as NSPRE. Aztec C is one example of a C compiler (and their assembler) which can use much longer names. My thanks to Morris Miller for this information. He has been teaching small C at our weekly Tucson Computer Group meetings.

I would like to be able to use even longer names, but again this is not a severe limitation.

Running the Pre-processor

The simplest way to run NSPRE is to type:

```
NSPRE Input file name.extent
```

When I did this to FBREAD.BAZ I got the output shown in Figure 2 on my terminal. The numeric variables are shown with the short variable name assigned to each one. The strings are done in the same way (note that FILENAME\$ was shortened to the eight character maximum, the numeric variables did not need to be shortened). OUTDEV was the first variable found and was assigned to A, FILENO was the second one I used and was assigned to B, etc. The order runs A-Z, A0-Z0,

```

rem Program FBREAD, sample single byte file reading program
rem for sequential or random access files
rem written by Saul G. Levy, Tucson, Arizona
rem
OUTDEV=0 \ rem output device #- 0=terminal, 1=serial, 2=parallel
FILENO=0 ; file #
rem
dim BYTE$(21),BYTE(21)
rem
@start input "File name,drive #: ",FILENAME$
FILETYPE=file(filename$)
if filetype<>-1 then @MAIN
! "File not on this diskette or drive number"
!
goto @start
@main open #FILENO%FILETYPE,FILENAME$,FILESIZE
@main2 for LOOP = 1 to 20
read #FILENO,&BYTE(LLOOP)
if BYTE(LLOOP)>31 and BYTE(LLOOP)<127 then BYTE$(LLOOP,LOOP)=chr$(BYTE(LLOOP
)) else BYTE$(LLOOP,LOOP)=" "
if BYTE(LLOOP)=1 then exit @print
next LOOP
@print for LOOP2 = 1 to LOOP-1
!#OUTDEV%3I,BYTE(LLOOP2),
next LOOP2
if BYTE(LLOOP2)=1 then !#OUTDEV," 1",
!#OUTDEV
for LOOP2 = 1 to LOOP-1
!#OUTDEV," ",BYTE$(LOOP2,LOOP2),
next LOOP2
!#OUTDEV
if BYTE(LLOOP)=1 then @eof
for LOOP2 = 1 to 21
BYTE(LLOOP2)=0
BYTE$(LOOP2,LOOP2)=" "
next LOOP2
goto @main2
@eof close #FILENO
end

```

Figure 1. The Program Source Code

NUMERIC VARIABLE ASSIGNMENTS					
A=OUTDEV	B=FILENO	C=BYTE	D=FILETYPE	E=LOOP	F=LOOP2
STRING VARIABLE ASSIGNMENTS					
A\$=BYTE\$	B\$=FILENAME\$				
LABEL TABLE					
40 @START	100 @MAIN	110 @MAIN2	160 @PRINT	310 @EOF	

Figure 2. The Assignment Table

```

10A=0\
20B=0
30DIM A$(21),C(21)
40INPUT"File name,drive #: ",B$
50D=FILE(B$)
60IFD<>-1THEN100
70!"File not on this diskette or drive number"
80!
90GOTO40
100OPEN#B%D,B$,FILESIZE
110FORF=1TO20
120READ#B,&C(E)
130IFC(E)>31ANDC(E)<127THEN A$(E,E)=CHR$(C(E))ELSE A$(E,E)=" "
140IFC(E)=1THENEXIT160
150NEXT E
160FORF=1TOE-1
170!#A%3I,C(F),
180NEXT F
190IFC(F)=1THEN!#A," 1",
200!#A
210FORF=1TOE-1
220!#A," ",A$(F,F),
230NEXT F
240!#A
250IFC(E)=1THEN310
260FORF=1TO21
270C(F)=0
280A$(F,F)=" "
290NEXT F
300GOTO110
310CLOSE#B
320END

```

Figure 3. The Original BASIC Program

```

10A=0 \ REM OUTPUT DEVICE #- 0=TERMINAL, 1=SERIAL, 2=PARALLEL
20B=0
30DIM A$(21),C(21)
40INPUT"File name,drive #: ",B$
50D=FILE(B$)
60IFD<>-1THEN100
70!"File not on this diskette or drive number"
80!
90GOTO40
100OPEN#B%D,B$,S
110FORF=1TO20
120READ#B,&C(E)
130IFC(E)>31ANDC(E)<127THEN A$(E,E)=CHR$(C(E))ELSE A$(E,E)=" "
140IFC(E)=1THENEXIT160
150NEXT E
160FORF=1TOE-1
170!#A%3I,C(F),
180NEXT F
190IFC(F)=1THEN!#A," 1",
200!#A
210FORF=1TOE-1
220!#A," ",A$(F,F),
230NEXT F
240!#A
250IFC(E)=1THEN310
260FORF=1TO21
270C(F)=0
280A$(F,F)=" "
290NEXT F
300GOTO110
310CLOSE#B
320END

```

Figure 4. The Edited BASIC Program

A1-Z1, ..., and A9-Z9. Note that FILESIZE was not assigned! Finally, the labels were assigned line numbers. Any reference to '@start' will GOTO 40. When an end-of-file is found we want to jump to the '@eof' label, so the program will GOTO 310 and finish up by closing the file and ending the program.

An output file called FBREAD.002 was produced as a CP/M version of a North Star BASIC (Type 2) program file which I tested with baZic (shown in Figure 3).

The complete command to run the pre-processor is of the form:

```
NSPRE Input file name.extent  
→ Output file name.extent CON:
```

The output file name can be any legal name (use .002 with baZic). 'CON:' is the output device and prints the assignment table on the console (the default). Use 'LST:' (LIST) for a hard copy. Note that the input and output file names must not be identical (the name or extent must have at least one character different). The default output file name is the same name as the input with an extent of .002.

The output device may also be a file name which will write the assignment table to that file. I have yet to get this option to work. If I give the old FBREAD.002 output file name and include a device file name (FBREAD.TAB), the old output file is not written into and a new file with the first few characters deleted is used instead (BREAD.002, EAD.002, or D.002)! The device file name is also created, but not written into! I tried various ways of using this option, but there must be a bug somewhere (with me, the documentation, or NSPRE).

The BASIC Program

Now look at Figure 3. The backslash in line 10 was from the multiple statement in this line (it was just before the rem in the source code). NSPRE expects you to use only a ';' or a rem WITHOUT the backslash (this varies from North Star usage).

Line 100 shows what happened to FILESIZE. NSPRE does not know about the very useful file size variable! I am

afraid that baZic is to blame for this (see below).

If you RUN this program with baZic (or North Star BASIC) you will get a SYNTAX error in line 10 (remove the backslash), then another in line 100 (change FILESIZE to S). The program will now RUN properly (see Figure 4 which is the version you should type in if you want to use FBREAD). Note that I reentered the rem that accompanied OUTDEV=0 back into line 10 of this version to aid you. Change the 0 to the printer port you are using when you want hard copy.

This points out that you can edit the BASIC program in any way you need to fix the errors that may occur. It is a BASIC program, after all.

All variables became upper case characters. The lower case prompt message in line 40 and the error message in line 70 were not changed. All of the rems were removed along with all of the unnecessary spaces. This was what we expected.

Note that it is very hard to figure out this program's logic with the variables assigned in alphabetical order and no REMs. The crunched format makes this problem even worse.

Of course this is good if you are selling your programs and do not want just anyone to figure out what you are doing. I prefer to nicely format my programs and use a lot of REMs to make clear what I am doing (about 60% of my programs are REMs and spaces!). It ends up that DIRCAT is already so large that it will have to be crunched to save time and memory during a RUN (I also alphabetized the variables which was a serious mistake). I expect to wear out the assignment tables while working on DIRCAT!

Also note what happened to the very long line in the FBREAD source code (it became line 130 in the BASIC program). The original line was so long that it would not fit on a single 80-column line. I used WordStar to load the source code, so I could not see this entire line at one time. This can also cause confusion since code which is hidden is harder to debug than code which is visible (making a paper listing is best, terminals still do not have enough lines on them!)

If you RUN the program, you will see something that looks like Figure 5. There are 20 bytes per line with the ASCII equivalent shown below each byte (line 130 makes sure that only the printable characters are shown). Line 190 prints the end-of-file byte (a 01H).

Is Any Program Perfect?

I mentioned above that all tabs are removed by NSPRE. I originally had a serious problem with the tabs I used between the label names starting in column 1 and the first tokenized key words in a number of those lines (a tab also started every other line). The offending tabs were not handled properly and some of the key words were not written to the FBREAD.002 file! I had to use a program which replaces all tabs with spaces to make NSPRE work properly (the columns still line up). This problem should be fixed. Also, Mr. Shearer did not use WordStar for the documentation (I sure hope he used a better editor!). The dot commands are not compatible with WordStar.

The missing file-size variable points out another problem: Which version of North Star BASIC is NSPRE written for? Mr. Shearer is using it exclusively (I assume) with baZic which defines the FILESIZE and FILEPTR function tokens differently than Release 5.2 BASIC does. This is not mentioned in the documentation, and means that NSPRE can incorrectly process programs written for use with Releases 5.2 or 2.1.1 BASIC (you will have to fix these by hand). It would be helpful to have this information in the documentation since I will be using NSPRE

to write programs only for North Star BASIC.

Mr. Shearer does mention that baZic's extended APPEND statement is not supported (both BASICs use the same token, but baZic can also APPEND code during a RUN!). I do not understand why NSPRE cannot handle the run-time APPEND statement as long as the line number option is ignored (the source code has no line numbers!). You cannot use APPEND within a North Star BASIC program because that BASIC uses APPEND in the direct-mode only. There may be other limitations hidden in the use of NSPRE that I have not found.

I would like to see these problems fixed by Mr. Shearer, or at least made clear in the documentation. I am very interested in using NSPRE, but am a bit leery of giving it the excellent recommendation it deserves (I need to spend more time working with it).

I applaud his efforts in providing us with such a useful program and would be happy to pay him for a legal copy of a corrected version (his copyright notice does not allow me to write this review!). Can we expect a similar pre-processor for Microsoft BASIC (or am I being overly optimistic)?

If you already have a good knowledge of a North Star-compatible BASIC and expect to work at learning the idiosyncrasies of this pre-processor, then I can recommend it to you. The long label and variable names are worth the effort. I have saved the best part for last: How many hundreds of dollars does NSPRE cost? You can get a copy for only \$15.00! Do buy it!

###

```

16 0 0 0 66128 0 0 0 66 88 0 0 0 65 3158 78 65 77
      B      B P      A      N A M
69 32 32 32 32 32 32 32 32 32 32 65 68 68 82 69 83 83 32
E      A D D R E S S
32 32 32 32 32 32 67 73 84 89 32 32 32 32 32 32 32 32
      C I T Y
32 32 83 84 65 84 69 32 32 32 32 32 32 32 32 32 32 32 32 32 90 73 80
      S T A T E      Z I P
32 67 79 68 69 32 32 32 32 32 32 32 80 72 79 78 69 32 78 85
      C O D E      P H O N E N U
77 66 69 82 32 32 78 79 84 69 83 32 49 32 32 32 32 32 32
M B E R      N O T E S 1
32 32 78 79 84 69 83 32 50 32 32 32 32 32 32 32 32 32 32 78 79 84
      N O T E S 2      N O T
69 83 32 51 32 32 32 32 32 32 32 32 32 82 69 67 79 82 68 32 78
E S 3      R E C O R D N
85 77 66 69 82 32 32 1
U M B E R

```

Figure 5. Sample BASIC Program Output

NSPRE: A Response

By Jim Shearer

Mr. Levy's review of my preprocessor, NSPRE, is well done and generally accurate. I appreciate his taking the time to do it and also his uncovering of a couple bugs that eluded me.

1. REMARKS following the statement separator " ". Mr Levy is correct that REM, rem, or ; must not be preceded by the statement separator. I have corrected the documentation to note that fact.

2. C/80, the C compiler used for this program, is a cut above small C. With the optional math package, it is nearly as complete an implementation of C as is available for micros. I recommend it without hesitation to anyone wanting to learn C.

3. The NSPRE.com file is deceptively large since it includes space for variable and label tables. Moreover, in my experience, object code files from C are usually at least as large as the source code file if they contain many calls to library functions.

4. The eight-character limit on variables and labels is not a limitation of C/80; it is my choice. Certainly the single letter variables of North Star BASIC leave something to be desired. But, speaking as someone who doesn't type well, the

GradePointsForSecondSemester

variables that writers of Pascal textbooks would have us use are equally unappealing. Eight characters seemed to me a satisfactory compromise.

5. If the Variable/Label Assignment Table is to be redirected to a disk file as opposed to the console or list device, the full command line injection must be given. In other words, the source file, the output file, and the Assignment Table file must all be explicitly named. For example,

```
NSPRE amortize.baz amortize.002 amortize.asn
```

would process the file "amortize.baz" writing the North Star program to file "amortize.002" and the Assignment Table to disk file "amortize.asn". The documentation has been revised to make this point clear and the bug preventing its use has been fixed.

6. I was unaware that baZic used tokens for FILESIZE and FILEPTR different from those used by North Star BASIC. I appreciate Mr. Levy's bringing that to my attention. It is regrettable that in a business where standardization is an endangered species such apparently uncalled for modifications occur. I have prepared a second version of NSPRE for anyone using a CP/M-based BASIC conforming to North Star's original token table as opposed to Micro Mike's.

7. The run-time programs are admittedly hard to read. Nevertheless, unlike CBASIC or other compiler run-time programs, they CAN be debugged. The Assignment Table (Mr. Levy's Figure 2) should be listed for just that purpose. Serious debugging will probably be done at the source-code level.

8. The bug relating to embedded tabs has been fixed.

9. Mr. Levy is correct; WordStar was not used for the documentation file on the disk. That file was prepared for use with TEXT, another excellent and reasonably-priced product from The Software Toolworks (the people who gave us C/80). Realizing that my membership in INSUA will probably be cancelled and my standing in the computer community in general seriously jeopardized, I must confess that I find WordStar a pain in the keester!

10. The run-time APPEND statement is not implemented because of variable-assignment considerations. If a program to be appended is written using "high" variables (e.g. A9 - Z9) and no references to variables in the main program, it CAN

be APPENDED. I've done just that with some common functions used in several other programs.

11. The MBASIC preprocessor (mbpre) source code appears the May Issue of Microsystems. I wrote it before NSPRE since it was much simpler. Now's your chance to learn C! Buy a copy of C/80, study the source code in that magazine, compile it and you'll have a working preprocessor for MBASIC.

After reading these comments, you may think I am an employee of The Software Toolworks. I have no affiliation at all with that Company; in fact, I don't know anyone who works there. It's just that I am deeply grateful to those people for providing the computing community with outstanding software at very affordable prices. I hope NSPRE, in a small way, exemplifies that philosophy.

#

Letters to Joe

INSUA:

A reader recently sent me a package asking for help in getting the TELSTAR modem program (INSUA disk #1003) going with his system. He also had a problem with my Two Column List routine (Compass, Vol. 3, no. 4, p. 25) It was printing two of everything.

Now, I usually pride myself on being lucky enough to find the answers to most such problems but this time I'm stopped cold. Why? Because nowhere in the 12 pages of documentation, or on his letter, or on the envelope, was his name, address, or any clue as to who sent it. The envelope was even delivered without postage!

I am more than willing to help when I have time, (I do work for a living!) but sometimes I need additional information or wish to reply directly to the writer.

I offer the following suggestions when corresponding with any author of articles in Compass:

1. Give as much information as you can. You may include printouts, memory dumps, schematics and anything else which may shed some light on the problem.

2. If you send a disk, use a SSDD format. I would guess that 99% of authors have DD but only about half have DS.

3. Include a printout of all correspondence. I can read WordStar files but that's about all. If your letter is on the disk prepared with Whizbang WordSmasher, it's going to stay on the disk!

4. If you wish a reply, or your disk returned, be sure to include a self addressed, stamped envelope. (SASE) Be sure to use a suitable mailer and include enough postage for disks.

5. Identify yourself. Anonymous letters give me the uncomfortable feeling that maybe I'm being asked to solve a critical problem for the Russian KGB or something.

And finally; if the person who sent me the package will write again, maybe I can help.

Joe Maguire
2321 Foxhall Drive
Anchorage, AK 99504

PIZZA CALCULATOR

SIZE	9 in.	11 in.	13 in.	15 in.	17 in.	18 in.	23 in.
IN^2	63.6	95.0	132.7	176.7	227.0	254.5	415.5
.050	3.18	4.75	6.64	8.84	11.35	12.72	20.77
.055	3.50	5.23	7.30	9.72	12.48	14.00	22.85
.060	3.82	5.70	7.96	10.60	13.62	15.27	24.93
.065	4.14	6.18	8.63	11.49	14.75	16.54	27.01
.070	4.45	6.65	9.29	12.37	15.89	17.81	29.08
.075	4.77	7.13	9.95	13.25	17.02	19.09	31.16
.080	5.09	7.60	10.62	14.14	18.16	20.36	33.24
.085	5.41	8.08	11.28	15.02	19.29	21.63	35.32
.090	5.73	8.55	11.95	15.90	20.43	22.90	37.39
.095	6.04	9.03	12.61	16.79	21.56	24.17	39.47
.100	6.36	9.50	13.27	17.67	22.70	25.45	41.55
.105	6.68	9.98	13.94	18.56	23.83	26.72	43.62
.110	7.00	10.45	14.60	19.44	24.97	27.99	45.70
.115	7.32	10.93	15.26	20.32	26.10	29.26	47.78
.120	7.63	11.40	15.93	21.21	27.24	30.54	49.86
.125	7.95	11.88	16.59	22.09	28.37	31.81	51.93
.130	8.27	12.35	17.26	22.97	29.51	33.08	54.01
.135	8.59	12.83	17.92	23.86	30.64	34.35	56.09
.140	8.91	13.30	18.58	24.74	31.78	35.63	58.17
.145	9.22	13.78	19.25	25.62	32.91	36.90	60.24
.150	9.54	14.25	19.91	26.51	34.05	38.17	62.32

Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

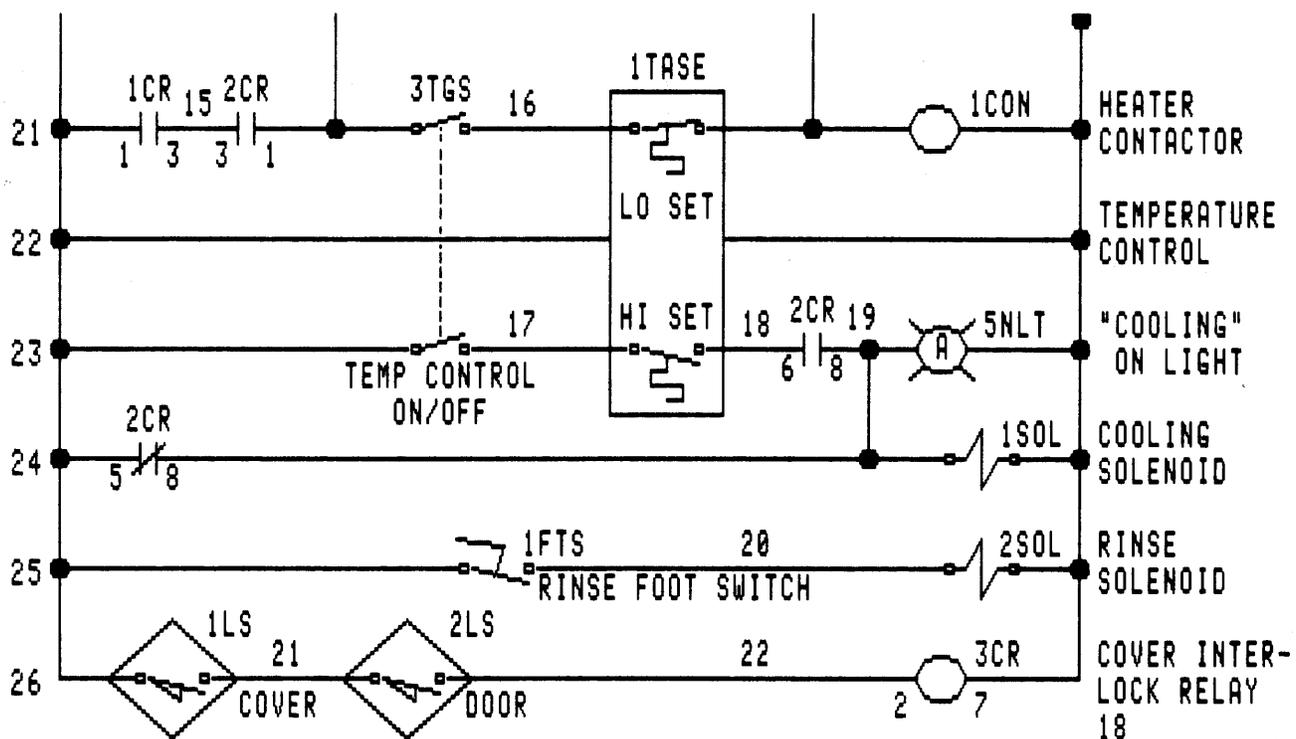
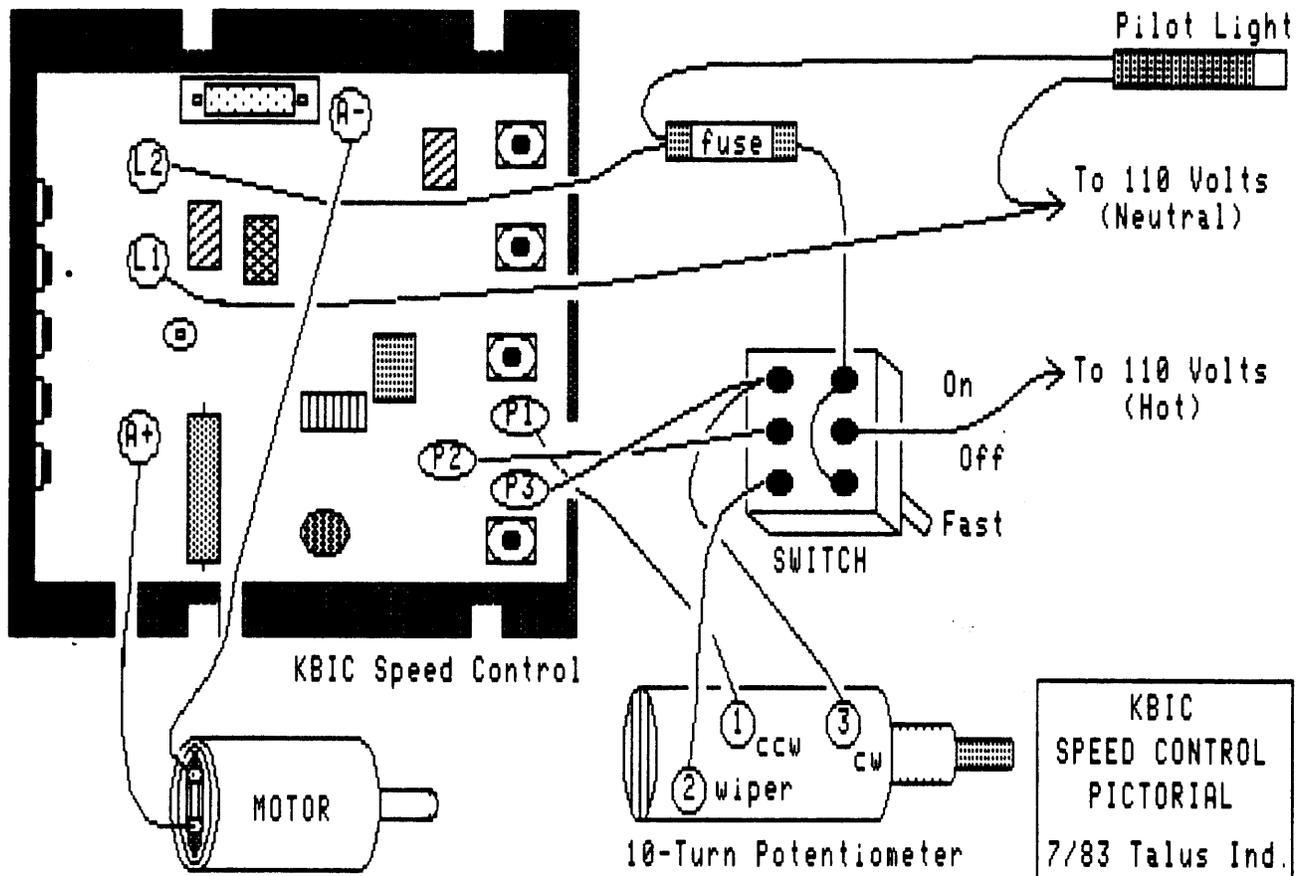
Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203

TATE 276H
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

The Compass

International NorthStar Users Association

Volume IV No. 3



Notes from the Editor

The gap between the present issue and the last one was greater than intended, but your editor was out of the country for two months on an academic research project, and could not get it out any earlier. INSUA intends to make up for this by publishing the next issue in less than two months. The issue will be devoted largely to members' concerns, and will contain, among other things, **a list of all current members, by name and zip code only.** We state this at the outset in order to give anyone who wishes his or her name not to be listed a chance to make that request by writing to INSUA at the normal address. **Please reply by return mail if you do not want your name listed!**

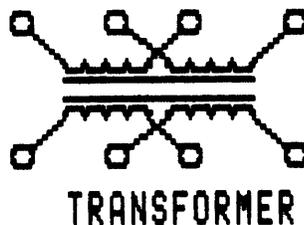
INSUA continues to maintain a policy of not selling or giving out its mailing list, even though several members have requested that this be done. The Board of Directors will consider soliciting the membership for advice on this matter at the time of the next membership renewal early in 1985. For the present we will once again print only names and zip codes so that any member who wishes to make the effort can contact North Star users and other INSUA members in their immediate vicinity.

INSUA is attempting to make improvements in two specific directions over the next several months. Bob Gary has contacted a number of volunteers who are sifting through whole libraries of public domain programs to cull out the little gems which have been written only to be forgotten about. **Any member who is interested in assisting in this software evaluation and disk library development program should write immediately to Bob Gary c/o the INSUA mail box.**

INSUA is aware of the letters from members which too often go unanswered. Although INSUA cannot deal with all individual inquiries, the Board of Directors wishes to improve its track record in this respect. **The Board therefore solicits volunteers who will agree to correspond with members' inquiries.** Kindly give us your name, address, and areas of expertise (or at least competence). Our plan is to publish the list, with an announced period of one year validity, suggesting that members direct their questions to the volunteer of their choice. The period of validity should help to insure that a member who joins in 1985 will know that he or she should not try to contact an expert who volunteers for the 1985-86 INSUA year. Volunteers should write to The Editor, c/o INSUA.

The August 1984 issue of Microsystems contains several articles on networking and, in particular, on TurboDOS. See "For Networks and Multi-User Systems--TurboDOS," by Ron Fowler; "The Networking Capabilities of TurboDOS," by Michel Simon and William Poole; and "TurboDOS spans the Horizon," by Karl Sterne of North Star. These articles should be of special interest to Horizon owners, for the Horizon, as Sterne notes, "is one of the few pre-1980 microcomputers still in demand." Startling, when you think of it, but wonderfully true!

Finally, our thanks to Steve Noll, whose graphics, produced with an Advantage and a dot matrix printer, grace this issue. Steve writes that he uses a NEC 8023 printer. His program is in GBASIC, and has file handling capabilities. He has created and maintains a library of electrical control schematic elements. A printout takes two to three minutes, high resolution graphics of course taking longer than low.



The Compass

The Compass is published every two months by INSUA, the Interational North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, Compass, P.O. Box 2910, Fairfield, CA 94533.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

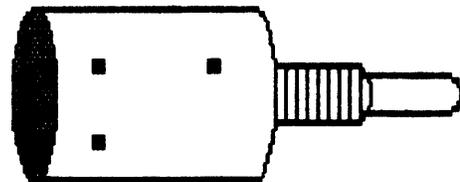
DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

Contents

- 2 John A. Bryant **A Bubble Disk on the Horizon**
Installing an S-100 Bubble Memory Board
- 5 David M. Allen **Graphics from Turbo Pascal**
A Turbo Pascal Program for Advantage Graphics
- 6 Randall Staponski **A Low Cost Word Processor**
A Review of and Introduction to SECRETARY
- 8 Saul G. Levy **Notes to a SECRETARY**
Supplement to the SECRETARY manual
- 21 Joe Maguire **Modifying NS S-100 Boards for
Extended Addressing**
Upgrading the Horizon for 64K-plus memory
- 24 Warren Lambert, editor **Alien Equipment**
Randall Staponski reviews Epson Buffer Boards
- 26 Steve Noll **Commentary**
Computer Insurance, SCAN & REN, Graphics
- 29 **Vendors Column**
Fischer Computer Systems, Angwin, CA
- 30 **GRAF LISTING**
Pascal Program Described by David Allen, p. 5
- 32 **Letter(s) to the Editor**



10-Turn 5K Ohm Potentiometer

A Bubble Disk on the Horizon

John A. Bryant
6648 N. Canandaigua Road
Holcomb, New York 14469

Having grown accustomed to totally in-memory word processors with their resultant speed and lack of irritating, frequent disk drive activity upon invoking various commands, when I finally bought Word Star I was rudely shocked. Since for years I had used only North Star DOS, I didn't realize how infernally slow CP/M was. When the two were put together--Word Star and CP/M--I came to a new realization of just how long a disk drive can churn and grind to save a simple little file.

I discovered that in Word Star some of the commands I use frequently are incompatible with each other; that is, they reside in different Word Star overlays, so that use of one command following the other requires accessing the disk and loading the second command. For example, when doing correspondence, the first thing I usually do is type the date, then use the ^O^C command to center it. If I am fast enough typing the date, then when I invoke ^O^C the disk will still be spinning from having created the file, but if not, centering requires the drive to turn on, then the overlay transferred to memory. Line delete, Y, lies in a different overlay, so when I decide to eliminate a line of text, the drive has to turn on, the overlay load, etc.

The slowness with which this all operates, and my reluctance to keep popping the drives on and off, resulted in my forgoing use of many of the Word Star commands. Instead of centering with ^O^C, I would use spaces and eyeball it, or I would repeatedly use word-delete (^T) instead of line-delete. While I did decide to continue to use Word Star in preference to my totally in-memory word processors, I nonetheless hated it because of its slowness and incessant disk activity. I thought of getting a ramdisk or disk emulator, but in addition to being expensive, that hardware would require me

to copy the three Word Star files into its memory before each use. The time involved with that procedure would probably outweigh the time saved.

The Answer

Obviously, the answer was a disk emulator using bubble memory. I spotted the two-part article in the January and February Byte magazines on building a bubble memory disk emulator for the S-100 bus and interfacing it to CP/M (and the Micropolis DOS) and decided to build when I found that the board could be built for about \$250.00 using wire wrap (which I had never done). For a price comparison, I have noted that Pure Data, Ltd. of Ontario, Canada, lists its 128K bubble memory disk emulator for the IBM PC at \$758.40.

The 128K bubble memory module from Intel was on special for \$199 when I bought mine; currently it sells for about \$340, I believe. It took, however, three months from the time I ordered the module through my local Hamilton-Avnet for it to come in.

In the meantime, quite by accident, I saw an article in the October, 1983 Microsystems on the same subject, along with CP/M and FORTH software. The circuit designs were slightly different in the two articles. I decided to follow the bubble memory module mounting technique and position from Byte, and the interface circuit from Microsystems. Parts other than the bubble memory module were readily acquired from Byte advertisers, along with an OK Tool wirewrap kit. (What a difference wirewrap makes over point-to-point soldering!!)

The device now works like a charm; I'm of course using it as I write this. If you decide to build it, and I recommend doing so, you should know the problems I had getting the thing up and running.

Construction and Interfacing Problems

There's a conflict in the two designs (Byte and Micro-systems). The Microsystems circuit has the RESET/ line (H on the bubble memory module) connected to 75 on the S-100 bus. This line is specifically discussed in the Byte article, with the conclusion that it should not be connected, but should be left floating. I left it floating and things seem to work fine.

As soon as I decided to build using the Microsystems design I wrote the author and told him that I was going to build his design and asked if there were any errors in the circuit as printed in the magazine. It has been my experience that there ALWAYS is at least one error. I pointed out that all of the pin numbers had been left off one of the ICs, but that I knew what they should be. Unfortunately, I received no answer. I say "unfortunately" because there was a serious error, one that took me many hours to track down: the same IC that had the missing pin numbers on the right side had the pin numbers on the left reversed from top to bottom. No wonder the board didn't work when I got it built.

That IC is not numbered in the circuit, but it is the 74LS244 labeled "IN". The correct pin numbers, from top to bottom, are:

```
18 :-----: 2
16 :       : 4
14 :       : 6
12 :       : 8
 9 :       : 11
 7 :       : 13
 5 :       : 15
 3 :-----: 17
```

One other error in the schematic is that the address buffer IC is labeled 74LS144; it should be a 74LS244 as well.

For some reason the schematic does not show any bypass capacitors. It probably would be a good idea to spread around some .1 mfd disc ceramic capacitors from the ground to the power pins on the ICs.

Software

Although the Microsystems CP/M software interface was flawless, I had two problems with it. The first was that I mistook a lowercase "L" for a one (shouldn't have if I had been thinking as I typed the program in), and the second was that I didn't know enough about CP/M to load and run a program assembled not at 0100H, but at 0FB00H. The CP/M LOAD command kept spitting it out when I used it on the .HEX file.

The software requires about 1K of memory space. If you are running North Star's standard PROM and have a full 64K of memory with the section where the PROM lies locked out, then you can use the Microsystems software without change; the software will lie above the PROM, out of harm's way. I, on the other hand, some time ago installed Bob Hogg's relocated PROM at FC00H (which I learned about through a Compass article), thus yielding a full usable 63K. I had to generate a 62K CP/M using CPMGEN, and relocate the Microsystems software to 0F900H and 0FA00H. That was no big deal, however; just change the two ORG statements.

The software was written with the bubble device as Drive C; I have a three drive system, so I changed the "bubdisk" EQU from 2 to 3, to set the device as Drive D.

With much thinking and reading I learned how to get the software to load and run at 0100H. It would be common knowledge, I expect, to anyone with good knowledge of CP/M, but I know only enough about CP/M to despise it. North Star DOS I know quite well, and think it is superb, as is North Star BASIC.

Anyway, the way to get the software to load and run at 0100H is to assemble it as usual, then look at the .PRN file and find out the numeric value of the "offset" EQU (equate) at the beginning of the program. Next follow these steps: (1) Log in to the disk containing the software's .HEX file. (2) Load DDT. (3) Use DDT's "I" command thus (the dash is the DDT prompt): -IFILENAME.HEX <CR> (where FILENAME is what you have named the software and <CR> is a carriage return). (4) Use DDT's "R" command to load the file to memory at 0100H by using

the offset value determined a moment ago by looking at the .PRN file: -Rnnnn <CR> (where nnnn is the numerical offset value). (5) Exit DDT with Control-C. (6) Use SAVE to save the memory image to disk as a .COM file: SAVE 3 FILENAME.COM <CR>. (6) The software is then executed by invoking it as any other CP/M software: from the $\frac{1}{2}$ prompt, type FILENAME <CR>. If all went right you'll be returned to the $\frac{1}{2}$ prompt and can then access the bubble disk with the various disk commands. I'd suggest a global erase (ERA *.* <CR>) right after logging into the bubble drive so as to clear what garbage it can from the directory.

A Problem with Garbage

I ran STAT on my new drive and was shocked to find that I had only 69K, not 128K. Using global ERA (ERA *.*) on the bubble disk did not help. Investigation revealed that for some reason CP/M thought many of the sectors of the disk were in use, and it could not be convinced otherwise by use of ERA. Apparently garbage in the directory area was reserving those sectors, but it could not be killed with ERA. The existence of these pseudo-files which would not print on a DIR listing was further inferentially confirmed by the fact that I could get the directory to hold only 20 files instead of the 32 it should have held. I think FORMAT would have solved the problem, but for some reason FORMAT does not work on the bubble disk. I could not think of any easy way to overwrite the directory area.

After much head-scratching, reading, and experimenting I finally cured the problem, and it did turn out to be a problem in the directory area. I can't vouch for how or why it worked, but what I did was revise the software to make CP/M think that the disk started with sector 3, thus ignoring the first few sectors. (Each sector of the bubble disk is 128 bytes.) This was done by changing the offset factor at 0FD4DH in the magazine listing from 0 to 3. I had previously used ERA *.* to clean out what it would of the directory, and after running the revised software I created

several files in the disk. As expected, STAT showed that I had much more memory than before because the garbage files in the first few sectors were skipped over.

I rebooted CP/M, then ran the original software. This had the effect of moving the beginning of the directory back to the beginning of the disk, but the directory still included the newly created files. Use of DIR revealed several non-printing files ahead of the ones I created. (They could be observed by the presence of several colons, used by CP/M to separate directory listings on the same line.) I used ERA *.* again, and lo and behold, STAT then revealed a full boat of memory.

Conclusion

I'm now much happier with Word Star, although it does now and then turn on the drives despite the fact that it is wholly located in the bubble disk. The commands that are brought in from different overlays appear to be in memory since they now operate quickly and without turning on the drives. Saving files to the bubble disk is much faster, and without the grinding and churning that used to occur before. Since Word Star and its overlays take up better than half the 127K available memory, I just keep current documents in the bubble memory. When I've finished with a document, I transfer it over to a floppy.

I haven't run any speed comparisons, though I observe quite a difference. Bubble memory is not as fast as RAM, by a good margin, but it is much faster than a mechanical disk. Now that I have solved the problems of the incorrect schematic, have got the software working, and solved the directory problem, I'm so pleased with the improvement that I'm thinking about building a second bubble memory board. Since it is port-addressed and requires only two ports there is no problem adding additional boards, so long as there is space enough on the motherboard. (A wire-wrapped version takes up the space for three boards, unless it is in the first or last slot, in which case it takes up the space for two.)

Obviously, the bubble disk is not limited to use with Word Star; you can do with it anything you would do with a regular disk drive, though it holds somewhat less. Use of Word Star in the bubble drive does have one significant limitation, however. Word Star (for no good reason that I can think of) insists on being on either the logged drive or drive A. Thus, if you put Word Star on the bubble disk, don't plan on logging into other drives. This is not as big a limitation as might appear at first glance, since reading and writing from and to other drives is permissible, using ^KR and ^KW. Anytime you really need to log onto another drive just boot up with a Word Star disk on drive A.

One of the nice features of the software in the Microsystems article is that no permanent changes are made in CP/M. Thus, you don't have to have two versions in case you want to run something without the bubble disk. When you do

want the bubble disk, just load and run the software from the .COM file. This patches in the jumps and writes the bubble disk routines, all in about one second.

Help Wanted

Ideally a Horizon owner who runs both CP/M and North Star DOS should have two bubble disks, especially if he has disk intensive software that he runs on both systems. I would like to build a second bubble memory disk for use with the North Star DOS, but do not know disk I/O well enough to write the routines. If anyone has done or does this, I'd appreciate hearing from him.

The Microsystems author suggested but did not implement what seems like a good use for the bubble disk: using it for a warm boot. If anyone has written or writes a patch to do that, please let me know.

#

Graphics from Turbo Pascal

David M. Allen
Department of Statistics
University of Kentucky
Lexington, KY 40506

North Star provides Advantage users who program in BASIC with easy-to-use procedures for graphics. Those programming in assembly language are provided with detailed instructions and examples. People who program with non-North Star compilers must improvise. This article is for Turbo Pascal users. It makes graphics with Turbo Pascal as easy as it is with North Star BASIC. (Turbo Pascal is a product of Borland International, 4807 Scotts Valley Drive, Scotts Valley, CA 95066.)

The Pascal procedures for Advantage graphics are nested in a program, GRAF, which is listed pp. 30-31. GRAF is driver program that makes it easy to exercise the procedures. I suggest that you copy the entire program, get some experience with it, and then copy the procedures into your own application programs. The listing should be in a CP/M file named GRAF.PAS.

To run GRAF in Turbo's memory mode, first execute GMGRADD with TURBO.COM in the command line. Then operate the Turbo system in the usual way. You might want to rename this version of Turbo Pascal to GTRUBO.COM and use it only for graphics applications.

To produce a command file that can be ported to other Advantages that don't have the Turbo system, do the following. Execute the Turbo system, use the options command to have the compiled program placed in a command (.COM) file, compile GRAF, and return to CP/M. Execute the program GMGRADD with GRAF.COM on the command line. GRAF is now ready to run. This method allows the use of much larger programs than does the memory method.

Notice that the procedures use Turbo Pascal's INLINE function where machine code is entered directly. To modify my code or to add your own procedures, the

machine code listings to the left of the assembly language listings in North Star's "Graphics CP/M Preface" are a big help. Variable names may be used to represent addresses where their values are stored. Literal constants are eight bits long, and Pascal constants are 16 bits long.

Pascal MT+ also has an `INLINE` function. The procedures given here can easily be modified to run with Pascal MT+. However, Pascal MT+ costs about eight times as much as Turbo Pascal, and I don't like it nearly as well.

`DOT` places a dot on the screen at coordinates (X,Y). For all procedures X and Y must satisfy $0 \leq X \leq XDOTS$ and $0 \leq Y \leq YDOTS$. `LINE` draws a line from the current position to coordinates (X,Y). The current position is set by the most recent call to `DOT` or `LINE`.

`SYMBOL` centers a symbol at coordinates (X,Y). `INITSYM` must be called before `SYMBOL`. `WCHAR` writes a

character so that the lower left portion of the character is at coordinates (X,Y). `WSTRING` writes a string at (X,Y).

`GRAF` is set up to run in the Advantage's hybrid mode. I think the hybrid mode is great feature. In applications where you want to use the full screen, `YDOTS` should be 239 instead of 199. The procedure `VIEWPORT` sets up the screen and must be called before any other graphics procedure. I frame the graphics area. If you don't like the frame, change \$81 to \$80. These procedures are set up so that the (0,0) coordinate is at the lower left corner of the screen. If you prefer (0,0) to be at the upper left corner of the screen, remove all statements "`Y:=YDOTS-Y`";. You will then need to adjust the "`Y:=Y-x`;" statements in `SYMBOL` and `WCHAR`.

If you distribute programs using these procedures or modifications of them, I ask only that you cite this article.

#

A Low Cost Word Processor

by Randall Staponski
3003 South Joplin Place
Tulsa, Ok 74114

Name:

SECRETARY

Type:

Word Processor

Hardware:

CP/M or HDOS

Price:

about \$5.00 from CP/M public domain libraries or HDOS version free to INSUA members who joined or renewed prior to June 15, 1984; \$15.00 from INSUA thereafter

After purchasing my North Star Advantage, I was dismayed to find that word processor packages ranged in cost from \$100 to \$500. This considerably dampened my desire to electronically pen letters to the editor, but did begin my quest for a less costly alternative. A chance request for a public domain CP/M catalog brought me in contact with the

CP/M version of SECRETARY. The HDOS version of SECRETARY available through INSUA will not run under North Star's disk operating system for the Advantage GDOS. However, the Advantage user who has CP/M can still run SECRETARY by using the public domain version. This review is nontechnical and is mainly intended for other beginners like myself.

SECRETARY is a first generation word processor (although it has undergone several revisions) originally written for the North Star Horizon and its operating system HDOS. The program was written when most microcomputers had 32K of memory or less. As a result it does not have all of the bells and whistles available on commercial word processors. The program has been converted to CP/M and placed in the public domain.

Although SECRETARY has a number of limitations as well as good points, it

should be adequate for the casual user. Serious users will probably find the program inadequate but can still benefit by determining which features are desired in a commercial word processor.

The program's most serious limitation is that it is line oriented. This means a line to be edited must be explicitly referred to and modified through the use of commands from the keyboard. SECRETARY does this by assigning each line in a document a number similar to a BASIC program.

For example, a displayed line might be:

10Laoks good

Assuming the desired line is "Looks good", the user would type

EDIT 10

The display would be

10Laoks good

The user would then use the usual North Star editor commands to change the line. For example, Ctl-A copies one character and moves the cursor one character to the right. Ctl-A would be pushed once followed by the correct letter o and finally Ctl-G to copy the remaining characters in the line. Additional control keys provide other editing functions. With the exception of Ctl-P, all of the editor control key functions available under GDOS BASIC appear to be the same as those used by SECRETARY. Line numbers and special print commands supplied by the user are displayed when editing a document. They are not shown when printing to the screen or a printer.

In contrast to the line editor, a commercial word processor is likely to be a full screen editor and does not display or require line numbers. The user simply moves the cursor through the text, usually via arrow keys, to the character(s) to be changed and makes the changes directly. In general a line editor requires more keystrokes during edit than a full screen editor.

Another source of irritation is the backspace function. In the configure mode, the manual indicates any key may

be chosen to backspace. On the Advantage the obvious choice is the destructive backspace key. Although a backspace does occur, it is not obvious to the user since the cursor **remains in place** on the screen. This worked okay if I only had to backspace 2 or 3 spaces. More backspaces than that caused me to lose count. As a result, I use Ctl-H (one of the built-in editor functions) for backspacing. It is non-destructive but does move the cursor one space back each time Ctl-H is pushed.

A 39 page manual plus an addendum for CP/M use is a part of the public domain package. It comes stored on disk. The manual often refers to the North Star disk operating system (albeit HDOS instead of GDOS) rather than CP/M. I actually found this preferable since I use GDOS more than CP/M. The use of examples makes the manual readable as well as usable. Also, I have found very few references to bits and bytes in the manual. This is quite refreshing since I do not even qualify for novice hacker.

SECRETARY has a number of good features usually found on commercial word processors. These include edit a line, load a file, save a file, indent, tab, underline, right justify, center lines, number pages, move one or more lines, copy one or more lines, find a string, and replace a string with another string. Frequently used paragraphs can be inserted using the chain command. Also long documents can be broken into smaller ones by the chain command.

Another useful feature is MERGE. This allows the user to insert data into a SECRETARY text file. One of the most common applications is to insert names and addresses into a letter that is to be sent to more than one person.

For example, a pointed letter to your Senator could also be printed with the correct name and address of your other Senator, your Representative, as well as your State Legislator. This is done by creating a data file with the desired names and addresses that are to be inserted into the letter. Whenever the character (caret) is encountered in the letter the next data string from the data file is inserted.

The data file containing the strings to

be inserted must not have any imbedded commands or characters. This means that SECRETARY, with its imbedded commands, cannot be used to create the data file. The manual suggests that a BASIC program be written to create the file. Conceivably a program could employ a considerable degree of decision making in creating the data file. However, for the names and addresses of my elected officials previously mentioned, I simply used the CP/M editor to create the file.

In summary, the user who normally produces documents a few pages long may find SECRETARY completely adequate. Even discriminating serious users who are

shopping for a word processor may find the program useful in determining the specific features necessary to meet their needs.

Finally, since SECRETARY is in the public domain it is available from a number of public domain libraries. I obtained my copy of SECRETARY from A & S at 849 Robinson Lane, Boulder City, Nevada 89005 for \$5. The \$5 fee includes postage and handling as well as the disk. There is no charge for the program. I obtained SECRETARY from A & S since they could provide public domain CP/M software on disks in North Star format.

#

Notes to a SECRETARY

By
Saul G. Levy
2555 E. Irvington Rd., Lot 47
Tucson, Arizona 85714
(602) 889-7753

Introduction

By now most of you should have the free INSUA MODEM7/SECRETARY diskette for early renewal of your membership in 1984. Let me run a poll of these members. How many of you loaded SECRETARY a few times, played with it, and stopped using it? I can see a number of hands raised! O.K. How many of you tried to use SECRETARY and gave up because you couldn't figure it out? Again I see quite a number of hands, which is not surprising!

SECRETARY is an excellent word processor which can handle most of our writing needs. It contains a fine selection of features without going overboard on those features you will rarely, or never need. SECRETARY was written by Gary Young who writes very interesting and useful software. These NOTES are intended to be most helpful to new users. Several serious bugs which still inhabit this program will be described along with additional information to help you.

A Little History

I first became acquainted with Mr. Young's software when I loaded some of his North Star BASIC programs in 1979. Every program I have seen of his has had bugs, often serious ones, and SECRETARY is no exception. I first saw and used SECRETARY at a friend's house later that same year. That version, even with its bugs, was very usable. Later on, another friend received a review copy and had a terrible time convincing Mr. Young that the problems were in the program and not in the hardware (several different Horizons showed the same problems!). I also used this version.

The main cause of these errors was the lack of a hard disk! Mr. Young had to write his source code in small modules. Each module was assembled separately, then the object code was combined into one program. Combining source code files is a tricky business and Mr. Young didn't do it correctly! After he got a hard disk, the major problems disappeared!

Last year the church I work for bought the latest version of SECRETARY (on my recommendation) to print form letters. That version is identical to the INSUA version.

I do not want my comments above to give anyone the wrong opinion of this program. SECRETARY is the best word processor available for North Star DOS, period! (There is also a version for use with CP/M.) Even the 1979 DOS version was great after using BASIC, or a BASIC program, as a word processor! It's well worth your while to spend some time learning to use this excellent program. I wish I knew enough assembly language to write something as useful!

Getting Started

This is not meant to be a tutorial on how to use SECRETARY (for a review see Mr. Ed Coudal's in Compass, Vol. I, No. 4, p. 24). The manual covers what you need to know. This section will tell you how to print the manual files if you haven't been able to figure that out yet. The manual is (almost) identical to the 1979 version! It is out of date (that's what the addendums are for) and contains a few poorly explained items. If you want to practice, you can always correct the typos in the manual! But first, you have to print it!

Start by making a working copy of the INSUA diskette. Never write on your master diskette! You could damage it or its files, and maybe end up voiding the warrantee (see What Warrantee? below)! Then boot your computer with Release 5.0 or 5.1 DOS (you're out of luck if you don't have one of them), load the working SECRETARY diskette in Drive 1, and type the following simple commands to the DOS:

```
GO NSSEC      Run either file (they are identical)
or
GO SECRETARY  Note the truncated spelling!
```

The following will be typed to SECRETARY:

```
N           Do not CONFIGURE the program for now (the # is
             the prompt for you to enter a command)
HELP        Lots of commands to choose from! Only the first
             four letters need to be typed (then hit
             RETURN). Read the manual before playing
             around!
LOAD TYPE-ME This is a small file which contains the
             non-warrantee! The program will give some data
             on the file it loaded. The space after the
             command LOAD is needed to separate any
             parameters which follow (the file name here)
LIST        List the loaded text file
QUIT       Return to the DOS (if you want to)
```

That is all there is to it! To load and print the manual you will have to CONFIGURE the program to use all of your available memory instead of the 7936 bytes free (this is done automatically when you configure). Rerun SECRETARY or type CONFIGURE at the # prompt. Answer the 9 or 10 questions in any reasonable manner for now (you need the manual to understand some of the questions!). If you save the configured version, it will be written to the SECRETARY file and that version will no longer ask if you want to configure when you run it. The NSSEC file remains unconfigured and is your backup (you also have the master diskette, don't you?). To print the manual, type:

```
LOAD USERMAN0 Load the first manual file
UNIT 0        Output port number: 0=terminal port, 1=right
              serial, 2=parallel (the same as with BASIC)
PRINT        Print on terminal for now
Control-C    If you want to stop printing
```

UNIT 0 is the default port and does not have to be typed above. The manual contains 35 pages, so it is wise to make sure that everything is O.K. before you start printing it. The other files will be loaded automatically. USERMAN5 is the addendum for the CP/M version of SECRETARY (ignore it for now). Be sure to load and print the North Star version's ADDENDUM file which will also print the ADDEND2 file. After reading all of this material you will have a better idea of how powerful SECRETARY is. Many of the four-letter commands are similar to BASIC's. You will be rapidly using them with just a little practice.

What Warrantee?

The TYPE-ME file contains INSUA's non-warrantee! I can't blame them for this statement. A users' group may even be legally untouchable for distributing this package which is now in the public domain (contact a lawyer if you want a legal opinion, it's your money). What gripes me is that many companies which sell software (often for mucho bucks) give you the same non-warrantee!

I don't know if Gary Young will continue to offer support for SECRETARY. He sold this package in exchange for the hard disk he is now using (or something to that effect). The program could use a bit more work! I also don't know who the buyer was (and INSUA has not told us where its copy came from). Now let me point out (and in one case, recover) some missing files!

Missing Files (and Other Confusion)

If you read the manual, you may have noticed that some of the files are missing! Figure 1 (below) shows the INSUA diskette's directory on the left and the church's on the right. The church's copy is in single density (the manual is on paper only).

ADDENDUM	4	30	D	2	SECRETARY	4	52	S	1	2D00	
ADDEND2	19	14	D	2	=SAMPLTTR	56	10	S	7		
DEMOFILE	26	16	D	2	SAMPDATA	66	5	S	3		
SAMPDATA	34	8	D	2	DEMOFILE	71	20	S	7		
USERMAN0	38	8	D	2	=LABELDOC	91	18	S	7		
USERMAN3	42	60	D	2	=BAS ICPGM	109	9	S	2		
USERMAN4	72	66	D	2	=LABELGEN	118	30	S	2		
USERMAN5	105	24	D	2	=LABELPRT	148	16	S	2		
USERMAN1	117	86	D	2	=LOADER	267	16	S	2		
USERMAN2	160	74	D	2	=SECR0E00	194	52	S	1	E00	
NSSEC	271	52	D	1	ADDENDUM	164	30	S	7		
SECRETARY	297	52	D	1	ADDEND2	246	21	S	7		
TYPE-ME	323	8	D	7							

Figure 1. INSUA's directory (on left), church's (on right with '=' signs before the missing files)

The seven missing files are:

SAMPLTTR	Sample form letter to be merged with SAMPDATA (a mailing list data file)
LABELDOC	Mailing label program documentation
BAS ICPGM	A simple program to write mailing list data files
LABELGEN	A complete mailing list program
LABELPRT	Prints LABELGEN's data files
LOADER	A SECRETARY loader which can tell which DOS you are using
SECR0E00	The E00H version of SECRETARY

I was very unhappy to find that the other version of SECRETARY was left out! I prefer to use Release 5.2 DOS at 100H which also gives SECRETARY an extra 7936 bytes of memory to fill with text. I don't like to switch DOSes all of the time! Note that the huge manual files didn't prevent the inclusion of all of the missing files (in the "hole" after USERMAN2)!

The SAMPLTTR file is actually on the INSUA diskette! I'll show how to retrieve it soon. The mailing list files are probably still owned by Mr. Young, and therefore not in the public domain. You don't really need the loader program.

I found two extra files on this diskette (in that big "hole")! They are complete copies of SECRETARY (sorry, 2D00H versions only). To retrieve them, jump to the DOS and type:

```
CR NSSEC2 52 219
TY NSSEC2 1 2D00
CR NSSEC3 52 245
TY NSSEC3 1 2D00
```

If you look closely at the file types, you will see that the text files are of Type 2. These are NOT BASIC program files! You can load and list some of them, but will see strange line numbers. An immediate SYNTAX error occurs if you try to run them. SECRETARY creates text files as Type 7's (it will load and save to any file type; loading assumes a special file format described in NOTE 12 below). Type 2 files cause a lot of confusion which we can fix. You should jump to the DOS and retype all of these files (except SAMPDATA) as text files:

```
TY ADDENDUM 7
TY ADDEND2 7
etc. through USERMAN2
```

Retrieving SAMPLTTR (or, What Happened to SAMPDATA?)

SAMPDATA should be a Type 3 BASIC data file containing one string record for each name and address on file. If you load and list this file with SECRETARY it will appear to be messed up, but note that it does contain the mailing list data followed by the sample letter text file! Ah ha! SAMPDATA contains two files, but the data portion is in CP/M format (a 1AH byte is the End-Of-File mark)! The text portion ends with the normal 01H byte that North Star BASIC also uses. Breaking up SAMPDATA is easy; converting the mailing list data to standard strings is a bit harder!

To break up the file, jump to the DOS and type:

```
DE SAMPDATA      Delete directory entry for this file
CR SAMPDATA 2 34 Create shorter file at same starting disk
                  address
TY SAMPDATA 3     Type as a BASIC data file
CR SAMPLTTR 6 35 Create text file one sector (two blocks)
                  higher on diskette
TY SAMPLTTR 7     Type it as a SECRETARY text file
```

Now we need a dummy file to hold the original SAMPDATA mailing list, then load and run Program CONVERT with BASIC. CONVERT will read the dummy file and write string output into the original SAMPDATA file (destroying the CP/M-formatted data). To do all of this, follow along at the DOS prompt:

```
CR DUMMY 2        Create the dummy data file
TY DUMMY 3        Type as a BASIC data file
LF SAMPDATA 9000 Load the CP/M formatted data into free memory
SF DUMMY 9000     Save it to the dummy file (use the same
```

GO BASIC
AUTO

address!)
Run BASIC to load Program CONVERT
Start automatic line numbering at line 10

Load CONVERT if it isn't on your diskette. The lines without line numbers are continued from the previous line and will fit. The listing shows some of the things that SECRETARY can't handle. The :s represent back slashes which SECRETARY would print as a form feed (BASIC will change them to back slashes). The ampersand is used for underlining so change it back to the & symbol in Line 110 (see NOTE 8).

```
10 REM PROGRAM CONVERT, CONVERT A SECRETARY CP/M DATA FILE TO
    NORTH STAR STRINGS
20 REM WRITTEN BY SAUL G. LEVY, TUCSON, AZ, JUNE 8, 1984
30 REM
40 DIM A$(165) : REM DIM OUTPUT STRING TO MAXIMUM SIZE
50 REM
60 OPEN #0,"DUMMY" : REM INPUT FILE IS A DUMMY COPY OF THE
    ORIGINAL FILE
70 OPEN #1,"SAMPDATA" : REM OUTPUT FILE IS THE ORIGINAL FILE
    NAME
80 REM
90 A$="" : REM NULL OUTPUT STRING
100 REM
110 READ #0,ampersandA : REM READ ONE BYTE FROM INPUT FILE
120 REM
130 REM CHECK FOR SPECIAL CHARACTERS
140 IF A=10 THEN 110 : REM LINE FEED (0AH)?, SKIP IT
150 IF A=13 THEN 220 : REM END OF STRING (0DH=CARRIAGE RETURN)?
160 IF A=26 THEN 260 : REM CP/M END-OF-FILE MARK (1AH)?, QUIT
170 IF A=34 THEN 110 : REM QUOTE MARK (22H=")?, SKIP IT
180 REM
190 A$=A$+CHR$(A) : REM ADD CHARACTER TO STRING
200 GOTO 110 : REM DO NEXT CHARACTER
210 REM
220 ! A$ : REM PRINT FINISHED STRING
230 WRITE #1,A$ : REM WRITE NORTH STAR STRING TO OUTPUT FILE
240 GOTO 90 : REM DO NEXT STRING
250 REM
260 CLOSE #0 : CLOSE #1 : REM CLOSE FILES
270 END
```

Save CONVERT if needed and run it:

```
NSAVE CONVERT 4 Save file in case we need it again
RUN Run it and you will see the five strings
which are written to the SAMPDATA file
(see Figure 2)
```

```
ABERT,STAN+123 STREET+CITY, CA 00000+++STAN++A00000
BAARY,MAN+STREET+CITY, ST ZIPCD+++MAN++AZIPCD
BAKER,TOM+STREET+CITY, ZIPCODE +++TOM++ACODE
FISHER,MARGE+ANDY+CITY, ZIPCODE +++CA++BCODE
POLISH+POLISH STREET+POLISH CITY, CA 99999+++NONE++C99999
```

Figure 2. The five new strings written to SAMPDATA (the +'s are field separators)

Now return to SECRETARY to try out the letter and data files:

BYE	Jump to the DOS
GO SECRETARY	Run SECRETARY
LOAD SAMPLTTR	Load the form letter
LIST	List the letter. The 's are where the mailing list fields will be merged (one per carat)
MERGE SAMPDATA	Merge the data file so SECRETARY knows that you are going to print form letters (one per name)
PRINT	Print them noting how the merged data is handled (read the letter too!)

SAMPDATA is not set up to print the salutation (Dear STAN, etc.). You can add another line to allow this: at the # prompt, type: 1201). Then merge again and print it (see NOTE 2). Note that the first and last names are not swapped into standard format. If everything looks O.K., you can delete the DUMMY file.

If you want to print the mailing list data on label forms, use a dummy letter which contains only 's and blank lines. Be sure to test this 'letter' and align the forms to suit.

Gary Young did publish Version 2 of his mailing list software (LABELGEN and LABELPRT) in Creative Computing, Nov.-Dec., 1978, p. 138. I have corrected errors only in this older version and have added three more fields for phone number, date, and notes. (Some of the original logic is quite obtuse!) The versions shown in Figure 1 are Version 3 files which include many changes, but still have errors. All of these programs are peculiar, run slowly, and can handle only small mailing lists (up to 100 names is reasonable). If you are a good enough programmer to make these programs work, then you should just write your own (SECRETARY accepts only the string format shown in Figure 2). Other mailing list file formats can be converted to what SECRETARY will accept. I have written such a program (called SECFOR) for use with MAILROOM files at the church.

A Useful Patch

After you have configured SECRETARY to use the rubout character that you prefer, you will find that the same character is echoed to the terminal. The input and output character must be the same. I want to input an underline (5FH) and output a Control-H (08H) so that the cursor will back up over the error (I dislike having to hit two keys for a Control-H when the key I prefer takes only one stroke). The only way SECRETARY will echo a Control-H is to input one as the rubout key. I tried correcting this problem, but SECRETARY kept replacing the Control-H with the underline! After several tries I found that the following patch will work.

You should add this patch only to a configured version! Load SECRETARY into free memory with the DOS and run one of the Monitors:

LF SECRETARY 2D00	Load SECRETARY at its normal address if you can (or use an offset: 3D00, 4D00, etc.)
GO MOE00	Run a Monitor which won't load over the copy of SECRETARY (a Release 5.2 version here)
DH 2D70,10	Display 16 bytes around 2D76-2D78 (add the offset if you used one: 3D70, 4D70, etc.)

You should see: 32 94 2E (which is a STA 2E94 instruction) at 2D76-2D78. These bytes have to be zeroed (a NOP instruction):

DS 2D76	Display and substitute command
---------	--------------------------------

32= enter 0 then space bar for next one (printed on this line)
94= enter 0 then space bar

2E= enter 0 then return to end

DH 2D70,10 Check that only these bytes have been
 changed to zeros!

DH 2E94 Display the byte where the echoed character
 is stored (CONFIGURE changes this byte!)

You should see: 08 which is the Control-H needed by my terminal. If it is not an 08, or you need another character, use the DS command to change this byte to the hex value your terminal needs. Check your work again with the DH command. If you make a mistake, it is far safer to start over again by reloading SECRETARY (you don't want to add other errors!). Exit the Monitor, save your patched version, and test it:

OS Jump to DOS
SF SECRETARY 2D00 Save it (add the offset if you used one)
GO SECRETARY Run it and make sure that the correct rubout
 character is echoed to the terminal (check
 the EDIT mode too)

If you get an E00H version, the corresponding addresses are: 0E76-0E78 (32 94 0F, a STA 0F94), and 0F94. What Gary Young needed this instruction for is unclear. It restores the output character when various commands are executed (SCRATCH is one of them). However, this character is never destroyed by other routines within the program!

Notes and Additions to the Manual

By using SECRETARY and learning more about it you will improve the way your writing looks. If you don't learn it, or don't care, your writing will show it. Take a look at the manual for an example (he wrote the program too)!

1. The useful pause character isn't explained in the manual, but it still works! The 1979 revision included the following at the end of Part 5 on p. 14:

P) PAUSING THE PRINT AND INSERTING DATA

The tilde character.....(7EH) can be inserted in the text to cause the printing to pause. Once paused, text entered at the console will be inserted into the line until a carriage (sic) return is entered. Control-C will abort the print. The line containing the new inserted text will not be justified. This can also be used to change print fonts -- pause the printing, change the type font, then enter a carriage return to continue.

A hand-drawn tilde occupied the ellipsis above because a real tilde in that location would pause for user input! To save you the trouble of knowing what to enter here, this part of the manual was deleted! To print a tilde at the pause: type a tilde, then a return to continue with that line which would have to be justified by hand. I put a COMMENT at the beginning of any file which expects user intervention during a print. The tilde allows you to print characters which SECRETARY reserves for formatting chores. It would've been nice if Mr. Young included a list of these (see Note 13 for my list).

2. If you stop a merged print with a Control-C, the merge function has not been turned off as the manual says. Typing just MERGE will allow you to enter the MERGE FILENAME command again to restart at the beginning of the merged file.

3. An error in the EDIT routine occurs when you are in the insert mode. If you use the rubout key, extra text from before the insert will appear within the inserted text. Hit

Control-G to copy the rest of the line and continue the edit (the line is still in the edit buffer, which is actually the input buffer). Note that this error doesn't occur in BASIC (they did it correctly).

4. The manual mentions that SECRETARY will accept control characters. This is nice, but these won't show in the files! If you embed them in a justified paragraph, the line which contains them will be printed too short (not justified). Also, your printer may overprint or print extra characters, and can go crazy when given various control codes. You can use the FIND command on control codes though you often won't know which one you typed! If FIND doesn't work, you will have to EDIT the lines around where the error is, typing Control-A's while you watch for a character which doesn't move the cursor (delete it). I hope your keyboard's switches are clean! Print again to make sure you have removed all of them. Using SECRETARY after a session with CP/M drives me crazy (Control-H's everywhere)!

5. I like to right justify almost everything. I also want two spaces (or more) to appear between sentences when I print. SECRETARY always inserts one space between the last word in a numbered line of text and the first word in the next line. I put two spaces between sentences which works fine except when the second sentence starts a new numbered line of text. SECRETARY adds the first space between the sentences, but may not add the second space when justifying the text it is going to print. The easiest way to fix the few places this occurs is to add a space between the line number and the beginning of the second sentence.

6. When you use the CONFIGURE or DESTROY commands, you must use upper-case letters only!

7. Sometimes I like to use a hanging indent like this example from my 11-page Hawaiian Itinerary:

Kaimu (Black Sand) Beach, one of the prettiest and best examples
of hot lava flowing into the ocean; bursting into sand; and
being washed back onto the shore, near Kalapana (KAH-EE-moo,
KAH-lah-PAH-nah)

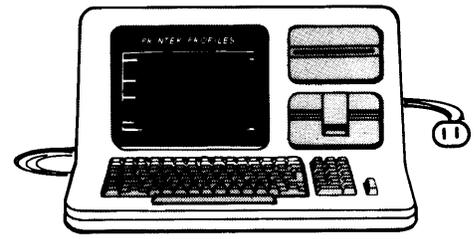
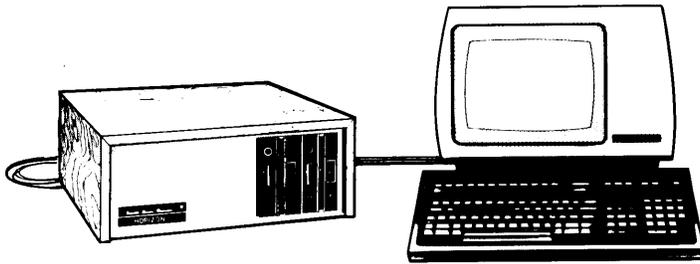
Note that this is not the full indent that the manual shows. I had to manually justify the first line, then indent the left margin, justify the rest of the text in the normal way, and return the left margin to its normal location before the next paragraph. This is a lot of extra work, but it does look good and allows quick scanning of the tremendous number of locations covered (each section of the four, main islands).

8. Underlining has the minor problem that you often have to include a punctuation mark in the underlined text (as the manual says). The ending & must be followed by a space. Contrary to the manual, when printing just an & sign, the & doesn't need to be preceded by a space if it starts a new line.

9. BLOAD and BSAVE are very handy features which didn't work very well in the past, especially on long programs (this is still a problem). Many BASIC programs will not print correctly (like Program CONVERT). Use LIST only (change UNIT numbers to list on your printer). BLOAD won't load a program which starts at line number 0 (text files can start at line number 0). I also wish that SECRETARY used five-digit line numbers so I could BLOAD my long BASIC programs!

10. After every file load, you should type RECOVER and check that the bytes used have not changed. This is the only way to check the files for proper format! RECOVER checks from the first line to the end. If the first line is messed up, the entire file is lost (unless you manually save it with a MONITOR)! RECOVER is always successful so this statement is misleading (it should be smart enough to save the good end to a damaged file). I can write a program to do this if anyone wants it.

PRICES SLASHED



APC's BASICs are New Standard in N★ "BLUE BINDER" Program

American Planning Corporation's two outstanding BASICs were among the first four software packages endorsed by North Star Computers, Inc., for inclusion in its prestigious "Blue Binder" program. Virtually all programs originally written in North Star BASIC will run directly under these two new BASICs. The American Planning Corporation BASICs are the new and even more powerful standard for North Star computer users everywhere. The APC BASICs are the only BASICs included, thus becoming the new de facto standard for North Star's Horizon and Advantage microcomputers.

The new Blue Binder program is a mechanism whereby North Star endorses third party vendor software for sale through its own distributor/dealer network. The endorsement is made after extensive independent testing and validation of the software by North Star. The company's faith in the third party vendor's product is evident in the fact that a traditional blue North Star three-ring binder is supplied to the software vendor for packaging of his programs.

Offered under the Blue Binder program will be:

MEGABASIC running under TurboDOS operating system for the Horizon with 8/16 board. Versions for other North Star machines are available under the "certified program."

APCBASIC which runs on the 8-bit machines in the line and is available under TurboDOS and GDOS.

Both BASICs support the full power of the North Star Advantage Graphics capabilities.

■ APC BASICs, TurboDOS Provide Speed, Power

The combined power and speed of the APC BASICs with the TurboDOS operating system provides much higher performance and user-friendliness than any machine competitive with North Star microcomputers. TurboDOS is a multi-user, multi-tasking operating system which operates a dedicated microprocessor and as much as 512 Kb of RAM for each of its up to eight users. There is minimal degradation in performance as additional users are added to the system.

Because TurboDOS emulates CP/M, a wide variety of application programs is immediately available to users. Under TurboDOS, the APC BASICs (APC BASIC for 8-bit machines and MEGABASIC for 16-bit computers) provide for automatic record locking when a record is read and automatic record unlocking when the record is written to. In special situations the automatic record locking feature can be overridden and fully controlled.

TurboDOS provides many sophisticated multi-user and networking facilities, using the record locking features, along with printer spooling and communications.

The structure of the operating system is distinctly modular. This allows creation of special versions for special needs. Thus the TurboDOS/APC BASICs combination is ideal for OEMs and system houses as well as multi-user and individual installations.

■ Real-Time High Resolution Graphics

"For a single-user CP/M system with high resolution graphics, the Advantage is just about unbeatable..." said a review of this self-contained, high-performance machine in an *InfoWorld* Report Card review.

The graphics capability of the North Star Advantage has been unmatched in the microcomputer industry since its introduction. APC BASIC provides extreme ease of use of the Graphics capability of the machine under applicable operating systems.

The Advantage, in addition to up to 256 Kb of main memory, carries an additional 20 Kb of 200-nanosecond RAM for use solely by the bit-mapped 12-inch green phosphor screen.

Graphics displays are presented in a high-resolution 240 x 640 pixel display. Basic geometric constructions such as circles, ellipses, lines and angles are generated by a quickly-learned and easy-to-use set of one-word statements/arguments in APC BASIC. Also controlled from the BASIC programs are a wide variety of shadings and fills to make the graphics even more vivid and meaningful.

Three dimensional graphs, line and bar charts, and even arcade-like games featuring independent motion by separate entities on the same screen are all easily within reach of a programmer using APC BASIC and the North Star Advantage graphics capability.

**AMERICAN
PLANNING
CORPORATION**

4600 Duke Street, Suite 425
Alexandria, VA 22304

INFORMATION/ORDERS: 1-800-368-2248
TECHNICAL HOT LINE: 1-703-751-2574

Dealer Inquiries Invited

Visa and Master Card Accepted

ON APC's BASICS

MEGABASIC

■ First Extendable Basic

The ability to extend the language and direct addressing of one megabyte of memory are two unique properties of MEGABASIC™, the new standard language for virtually all popular microcomputers using a 16-bit central processing unit. MEGABASIC achieves extendability by accessing "packages" from memory in much the same way that FORTH and a few other high-level languages provide for extensions of basic capabilities.

The package concept allows MEGABASIC users to load libraries of programs, subroutines and functions into memory that can be executed from the running program with a single statement as if they were an integral part of the MEGABASIC language itself. The packages are user-written in MEGABASIC, providing each user of MEGABASIC with the ability to tailor the language specifically to his needs.

With direct addressing of 1 megabyte of memory, MEGABASIC provides users of TurboDOS, MS-DOS, CP/M-86, and MP/M-86 operating systems full and flexible use of the power of IBM-PC microcomputers and similar machines.

Other features of MEGABASIC include:

- Execution 2 to 15 times faster than any other BASIC and faster than many compilers.
- Advanced TRACE and EDIT functions that allow easy program development and debugging.
- VERSATILE constructs allow use in business applications such as payroll and financial programming and in number crunching scientific and engineering programs.
- BCD floating point arithmetic to eliminate rounding errors.

APC's standard software package includes a RUNTIME semi-compiler version that further speeds execution by 30 to 50%. The software also includes a configuration program, compaction program which removes spaces and REMarks, combines short lines and scrambles source code for security, and a cross-reference program that provides listings of all variables, strings, arrays, subroutines, functions and other program information.

The 302-page reference manual contains total documentation for all features. The manual contains a 28-page (two columns to a page) index with more than 2,900 entries to pinpoint a user's need, no matter how defined.

APC BASIC

■ The 'Mercedes-Benz' of BASICs

APCBASIC, the forerunner of MEGABASIC, was called the "Mercedes-Benz of BASICs" in a *Microsystems* magazine review. It runs 2 to 6 times faster than other BASIC interpreters. It goes far beyond the proposed ANSI standard BASIC and all other microprocessor BASICs in providing users and programmers with a fully-featured powerful language for development work and applications.

The language was originally developed for use in complex government budgeting applications. Because of this, it provides features needed by advanced programmers while retaining the friendliness and ease of use of a limited BASIC for beginning users.

Among the features of this powerful language, which was field-tested for three years before being introduced, are:

- VERSATILITY. The rich constructs of APCBASIC allow its use in everyday applications such as payroll and financial programming and in number crunching scientific and engineering programs.
- EASE OF DEVELOPMENT. Program development is significantly easier because of APCBASIC's rich editing and tracing facilities, which are comparable to those found in mainframe languages. The editor enables a programmer to modify every occurrence of a variable or string, globally, by program region, or individually. Programmers can maintain a single program source that will run under virtually any operating system.
- PORTABLE. Programs written in APCBASIC are easily portable from machine to machine. Since there are versions of APCBASIC available for virtually all business machine microprocessors, such as the Z80 and 8086/88, and it is at home with most operating systems, including TurboDOS, CP/M-80, CP/M-86, MP/M, MP/M-86, PC-DOS, MS-DOS, and others, programs written for one machine and operating system can be moved easily and quickly to other machines and operating systems.

—FULLY FEATURED. Among the features of APCBASIC are 50 functions, 18 commands, and 42 statements, exact BCD arithmetic which eliminates rounding errors, an extensive function library with fast sorts, ready-made yes-no prompt routines, matrix inversions, and many other subroutines, and a runtime version which further speeds execution by 30 to 50%. The software also includes a configuration program, compaction program which removes spaces and REMarks, combines short lines and scrambles the code for security.

NEW PRICING... SAVE \$151 ON APC BASIC!

To make the new NORTH STAR "STANDARD BASICs" available to as many readers of *Compass* as possible, APC has cut the prices dramatically.

NORTH STAR DOS APCBASIC	\$249	(formerly \$400)
NORTH STAR CP/M APCBASIC	\$249	(formerly \$400)
NORTH STAR HDOS APCBASIC	\$299	(formerly \$400)
GRAPHICS DOS APCBASIC	\$299	(formerly \$400)
GRAPHICS CP/M APCBASIC	\$299	(formerly \$400)
CP/M-86 MEGABASIC OR APCBASIC	\$375	
MS-DOS MEGABASIC	\$375	
TURBODOS MEGABASIC OR APCBASIC	\$495	

These new low prices include the full package of languages and utilities including the DEVELOPMENTAL version of the language, the RUNTIME semi-compiler, a CONFIGURATION program, a COMPACTION program, a CROSS-REFERENCE GENERATOR and a 302-page REFERENCE MANUAL.

PRICE PERFORMANCE COMPARISON:

Product	Price	Performance*	Performance/Price
BAZIC	\$150	52	.35
APCBASIC	249	251	1.00

*Percentage improvement over North Star BASIC on five sorting benchmarks.

B.O.S.S. is a trademark of American Planning Corporation
 PCNet is a trademark of Orchid Technology
 CP/M & MP/M are trademarks of Digital Research, Inc.
 IBM PC is a trademark of IBM Corporation
 TurboDOS is a trademark of Software 2000, Inc.
 dBase II is a trademark of Ashton-Tate
 MS-DOS is a trademark of Microsoft

**AMERICAN
 PLANNING
 CORPORATION**

4600 Duke Street, Suite 425
 Alexandria, VA 22304

INFORMATION/ORDERS: 1-800-368-2248
 TECHNICAL HOT LINE: 1-703-751-2574

Dealer Inquiries Invited

Visa and Master Card Accepted

11. Saving your text on a diskette is an important operation, right? I don't mean to scare anyone, but sooner or later SECRETARY will trash a file, and usually, it will not be the file just saved (see **Two Nasty Errors** below)! At least this doesn't occur very often. If the text you are working on is important to you, always make another copy on a different diskette; otherwise, your hard work can be lost in a few seconds! This is even worse if you don't have a printed copy. Manual recovery is a tedious process to do correctly. You should keep the backup file at least until you are finished editing and have printed a final copy, but be sure to read **Two Nasty Errors** below!

12. Load a file and type: DUMP 60FF,612E then (UNIT) 0 to see what a SECRETARY file looks like (the lines are too long to fit in 80 columns). If this is not clear, run a Monitor and type: DA 60FF,40. Each line consists of:

Byte #

- 1 Hex number of bytes in this line including itself and a carriage return (0DH)
- 2-5 Line number in four ASCII hex bytes
- 6-... Text in ASCII hex bytes (132 maximum)
- Last Hex carriage return (0DH)

The address 60FFH is where each file starts loading. The last line will be followed by a 01H End-Of-File mark. This format started with Processor Tech and is called SCS (Self-Contained System) format. If you load a file which isn't in this exact format (BLOAD works differently), you will not be able to RECOVER all, or any, of this file. Note that SCS format is not what BASIC uses. BASIC tokenizes the key words and uses a two-byte line number in standard assembly format (low byte, high byte). BASIC can read and write SECRETARY text files only one byte at a time (slow!), or use the DCOM utility to read a whole file at one time (fast!). I have written a number of programs which use these files (see **Help is Available!** below).

13. Here is my list of reserved formatting characters (it belongs on p. 23 of the manual):

- # (23H) to imbed commands in the text
- Ampersands (26H) to underline the included text
- Less than (3CH) to center the following text line
- Greater than (3EH) to reset margins while printing
- Left (5BH) and right (5DH) square brackets to justify the included text (if justify is on)
- Back slash (5CH) to eject the paper to the start of the next page (top of form)
- (5EH) to merge a field of data from a data file
- Tilde (7EH) to pause during a print for user input, end with a return

Note that # is only reserved when it occurs in column 1 of a numbered text line and that is reserved only when a merge is in effect.

14. The CALL command does work, but whatever is at 2D33H won't! See p. 16 of the manual for what this is supposed to do.

Two Nasty Errors!

1. The first error is only nasty if you don't know how to prevent it from printing garbage all over your screen. If you spell a command wrong, the HELP list is printed (don't forget the space before any parameters entered!) If you make a mistake on the parameters, SECRETARY doesn't always just print an error message! I have noticed this when using COPY and MOVE. If you only enter one or two of the three parameters needed, SECRETARY will ask you for the UNIT number! Don't enter one, hit Control-C

instead! You may destroy parts of SECRETARY while trying to get the # prompt back (the program pointers have not been set for whatever the program is doing)! Sometimes the garbage it prints looks like the DUMP command's output (only DUMP asks for the UNIT number); other times it looks like you are trying to print an object code (Type 1) file! Why SECRETARY ends up in the DUMP (or whatever) routine, only Gary Young knows! I also managed to destroy the BLOAD routine while testing the INSUA version (the disk was never accessed)! If there is any question about the copy you are using, especially after SECRETARY acts strange, run it again or reboot.

2. SECRETARY has a serious bug when you save a (usually) large file. If your text will almost fill its disk file, SECRETARY will sometimes save an extra sector of memory and clobber any file which follows the one you are saving! SECRETARY can't RECOVER a text file clobbered in this way! Be aware of this when you edit old files. You should check that any file you save can be loaded and recovered properly (or you may lose it). Load the next file, if there is one, and RECOVER it too or you may get a surprise the next time you try to load it. I can't say how often this occurs or what file sizes won't cause this problem, but it did occur when saving a 26 block file which had four bytes free after the End-Of-File mark! I do know that NSAVE always seems to work (DESTROY and NSAVE is safer than just a SAVE!) Apparently, Mr. Young always uses NSAVE which adds four blocks to whatever is currently needed to save the actual memory used. What is worse is that SECRETARY saves memory equal to the entire file's space on the diskette, not just the amount of memory used! The review copy I mentioned before had an even worse error: SAVE wouldn't work on double-density diskettes (single density worked fine)! The current version has this problem fixed (or maybe only most of it was fixed!). The review copy also had other nasty errors!

Help is Available!

If you have questions or need help with SECRETARY, please contact me. Maybe we can start a SECRETARY users' group! This is not as romantic as INSUA, but if anyone is interested--why not?

I have been writing a number of BASIC programs for use with SECRETARY files. Others can be written at a reasonable cost. Here is what is available:

Mailing list package. SECRETARY text files make a simple and quick mailing list. The current version can handle only the last two digits of a five-digit Zip code, but I will add for free whatever is needed to use full five- or nine-digit Zips. To sort the Zips you will have to buy a sort package (ask for details). Cost: \$30.00.

CRUNCH Program. After editing a file you will end up with some lines too long and some with only a word or two; this looks messy if you send copies to the Compass. So I run my files once (and sometimes twice) through CRUNCH to equalize the amount of text on each line. A special option is included to NOT crunch unjustified tables and text (any special formatting would be ruined). CRUNCH will save you some line numbers and a few bytes of memory, but it doesn't shorten the file sizes used! I have been using a compiled version which runs faster than I can read the text (ask for details). CRUNCH can also handle some of the reserved formatting characters and can list SECRETARY text files. Cost: \$15.00 for both the BASIC and compiled versions.

QUIKFORMatter Program. I prefer to include a documentation file with my software so each user can print his own copies instead of getting them from me. I can't expect that everyone has a copy of SECRETARY, so I format a data file with unjustified text in standard BASIC strings and include that file on the diskette. Centered text will be centered and a title string can be entered so the package name and copyright notice (or whatever) can be printed on every page. Cost: \$30.00.

Quick print (QUIK!) Program (often called RUNTHIS), used to print the output of QUIKFORM above. Any length printer or terminal page can be used. Cost: Free with Program QUIKFORM above or one of my other packages.

READSCS Program, written by Randy Reitz. It will print or list a SECRETARY text file slowly enough to read it. Cost: Free.

Save \$15.00, order all of the above programs for \$60.00!

The Hawaiian Itinerary (actually, a guide) is also available for \$10.00. That funny thing on the end of the example is my stab at a pronunciation guide. The full listing would take a normal person about 10 weeks to see everything listed!

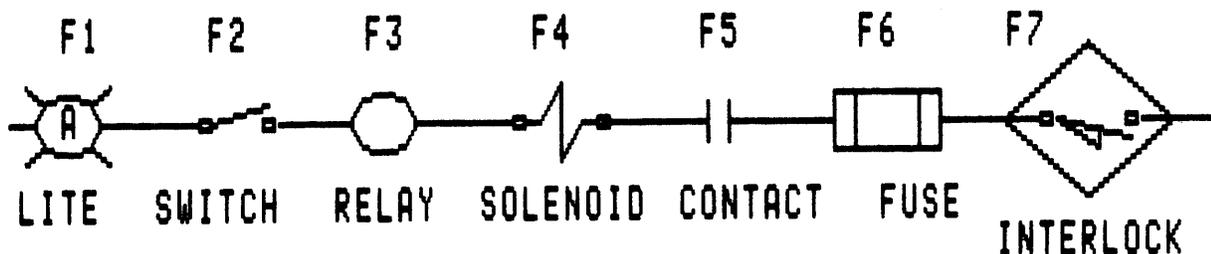
Most of these programs will let you stop the terminal listing by hitting the space bar (the code to do this must be changed for each terminal). CRUNCH and READSCS can also list any SCS file. If you have the PDS assembler, Program ASMB files are in SCS format (MAKRO files aren't).

If you wish to order any of my programs, send the amount listed plus \$3.00 for postage and handling on each order (not on each program) and include a well-protected diskette initialized in the density you want. If you tell me the density you want, you can order one of my diskettes for an additional \$5.00 (it is cheaper to use your own). Call or write for details (I need to know what terminal you are using and how much memory you have: the more the better). My warrantee is: Your money back within 90 days if I can't satisfy your needs. My programs are tested and will do whatever I claim for them (limitations are fully described). I provide full service and support with my low-cost software.

I hope that INSUA will also add extra files to the SECRETARY diskette that it is selling. I also wish that someone would clear up why there are missing files, and that (at least) the E00H version of SECRETARY be added to the INSUA diskette. If INSUA won't provide that version, and if it is in the public domain, I will provide it at no charge (except for the same postage and handling, and diskette fees, listed above). This also applies to this article which is available at no charge on a diskette (a paper copy is \$2.00 extra).

I have been rather critical of Gary Young in these notes. Even so, I have grown very fond of his secretary and will continue to use it for a long time to come (even if North Star DOS disappears in the next few years due to TurboDOS and the Dimension Computer). Someday I would like to meet Mr. Young; I believe we have a lot to talk about!

###



Modifying NS S-100 Boards for Extended Addressing

By Joe Maguire

If you are thinking about upgrading your system to use one of the multi-CPU boards, you have no doubt considered the memory problem. A CPU such as the 8088 can directly address up to one megabyte of RAM. Does that mean your older North Star memory boards are obsolete? Not at all. What follows is a description of how to allow your old boards to respond to extended addressing.

The 32K RAM Board

Modifying the 32K RAM board for extended addressing is simplicity itself. If you were one who modified your boards to 64K according to the article by Steve Leibson (*Compass*, Vol. II, no. 4, p. 6), you probably already have a spare handy.

Refer to sheet 3 of the schematic diagram for the 32K RAM. Notice that the IC at location 6B (74LS75) has one latch that is not used. That latch will do the trick.

Step one is to decide which address range you want the board to respond to. With this modification you have two choices: the first 64K (0000-FFFF) or the second 64K (10000-1FFFF).

Step two is the modification. Jumper pin 15 on the S-100 bus connector to pin 3 of IC 6B. Address line 16 (the first bit of the extended address lines) has been assigned to S-100 bus pin 16. Wirewrap wire can be used for the jumper.

If pin 16 is missing on your board, try to obtain some "stick-on" pads from any source which sells circuit design aids. Or, you can possibly use a sliver of the sticky metallic tape sold in hardware stores for use on furnace ducts and the like.

The next part of the modification is to connect one of the outputs of the latch to pin 4 of the decoder IC at location 6A. The cleanest way to do this, since pin 4 is connected to ground, is to lift the IC out of the socket, bend pin 4 out, and re-

insert the IC in the socket. If you want the board to respond to the first 64K, jumper pin 15 of IC 6B to pin 4 of IC 6A. If you want the board to respond to the second 64K, then jumper pin 14 of IC 6B to pin 4 of IC 6A. Only one of pins 14 or 15 is connected. The other is left unconnected.

Either solder the jumper to the pins or use push-on connectors salvaged from an old socket.

The 16K RAM Board

The 16K RAM boards are only slightly more difficult to modify than the 32K boards. That is because there is no spare latch on the 16K boards. The extended address line must be jumpered directly to the address decoder, IC 6A.

The earliest version of North Star's 16K RAM had no latch IC. The address lines went directly to IC 6A. Later versions did use a latch, probably because of some problems encountered with computers other than the Horizon.

Again, the determination must be made whether the board is to respond in the first 64K or the second. If it is to be the first 64K, jumper A16 to pin 4 of IC 6A after bending pin 4 out of the socket.

If it is to respond in the second 64K, jumper A16 to pin 6 of IC 6A. Pin 6 should be bent out of the socket in order to disable it from the phantom circuit to which it is normally connected. Phantom will not be available if pin 6 is used. Pin 4 should remain in the socket connected as normal.

Address selection is as usual, mentally adding 64K to all address blocks.

How it Works

When a CPU such as the 8088 is working in the address range between zero and 128K, it will actively use only 17

address lines. The remaining lines will stay at a logic low. This modification connects the 17th line (A16) to one of the chip enable pins of the address decoder. (74LS138) Unless pin 4 of this IC is at zero volts, and pin 6 is at +5 volts, the IC will not respond to any address.

If the CPU is addressing a location in the first 64K, line A16 is a logic low. This low is passed through the latch IC at 6B, (in the case of the 32K RAM) out pin 15 to pin 4 of IC 6A. IC 6A then responds to the memory requests as it always did before.

If the CPU does a memory access in the second 64K, line A16 will change to a logic high. This high is passed through the latch to pin 4 of IC 6A. Pin 4 is now at or near +5V and the IC will not allow any response from the board.

While all this is going on, pin 14 of IC 6B has just the opposite voltage level of pin 15. If it is connected to the other memory board, that board will respond in just the opposite fashion to the one just described. You now have 128K of available memory.

In the case of the 16K RAM board, the high on A16 is passed to the pin formerly connected to phantom. The board believes it is being disabled by the phantom signal and is now enabled again. For our purposes, the result is the same.

The Disk Controller

The DD disk controller board can also be modified for extended addressing although not as easily as the RAM boards.

The board select PROM, (DSEL) IC 11C, has two chip enable pins, 13 and 14. Both of these pins must be grounded before the board will recognize an address.

By using the spare inverter on the board (IC 14B pins 1 and 2) A16 could be jumpered to either pin 13 or 14 (through the inverter) to make the board respond to an address only in the second 64K. But this is not the end of the problem.

The disk program PROM (DPGM), IC 9D, is programmed in 8080 code. If you try to read this with an 8088, it's going to enter the twilight zone. Fast! That means the PROM must be read with an 8080, 8085 or Z80 before you can use the disk. If it's in the second 64K address

range, you must use some sort of bank switching scheme in order to do it. It can be done, and I will cover how in greater detail in another article, but it is not trivial. Caveat!

(A study of the disk controller schematic seems to indicate that the PROM program does not need to be board resident. That means that it could be transferred to RAM or ROM outside the controller and, if so, converted to 8088 code. For that matter, the whole DOS could be in 8088 code. But I'm going to leave that project to another author!)

Caution!

Once your memory boards have been modified for extended addressing, they should not be used in the computer unless there is something connected to the extended address lines. Either a 16 bit processor or a memory manager will do. Unless such a device is present, the lines may float causing erratic memory responses.

128K and up!

What happens if the CPU tries to address memory above 128K? The memory boards will become confused, that's what. In order to use the memory in a larger-than-128K system, more address lines must be decoded. There are still two more chip enable pins remaining on IC 6A but, if these are used, other functions, such as phantom or bank selection, must be given up.

Another alternative would be to treat pairs of boards as 128K banks and bank switch them.

Why would I want to do anything as crazy as that when I can buy a 256K or 512K memory board? Price--that's why.

Have you seen the price of 256K boards? I predict that the older generation boards will become available for next to nothing. If you are a dyed-in-the-wool hardware and software hacker, you will be able to assemble your dream system for peanuts. Besides, I'm waiting for Steve Leibson to tell us how to use 64K or 256K chips on our North Star 32K RAM boards!

The Pin 61 Problem

If you are planning to go beyond the 20 address lines of the 8088 and into the realm of full 24 bit addressing, you are going to run smack into the pin 61 problem of the Horizon computer's S-100 bus.

When North Star designed the Horizon in 1976, they thoughtfully added some extra ground pins. Of the three that they chose they got two hits and one strike. Pins 20 and 70 were adopted by the IEEE S-100 standards committee as grounds but pin 61 was selected for address line 20.

That wouldn't have been so bad except for the way North Star chose to connect pin 61 to ground. Pin 61, on the Horizon motherboard, is connected to the wide ground plane on the component side of the board under the sockets! This makes for good grounding but tough to undo. There are two possible ways to correct the problem.

The Simple Solution

The easiest way to solve the problem is to isolate pin 61 on the CPU board and the memory boards by sticking thin strips of paper over their fingers on the S-100 boards. Then jumper directly between boards with wirewrap wire.

This method works OK if only a few boards will be involved; but what if you plan extensive use of slave boards? Then the permanent fix should be considered.

The Permanent Solution

This involves some careful shop work. The motherboard must be removed from the computer and the ground connections of pins 61 drilled out from the solder side of the board.

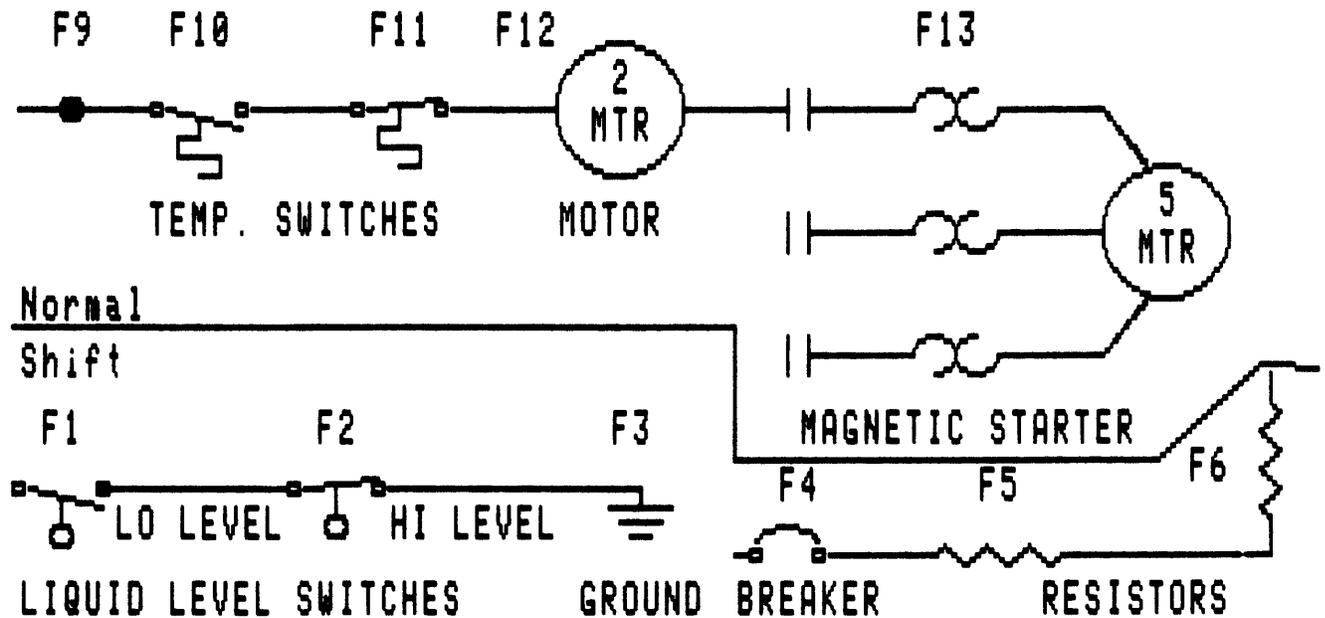
A hollow drill is required which can be made from small steel tubing obtained from hobby or model stores. Cut off about two inches of tubing and carefully notch one end with a file to produce some cutting flukes. Next, chuck the drill into a suitable drive. A drill press is best but a hand drill can be used if you use care.

The motherboard should have been prepared by removing as much solder from the pin 61 holes as possible. Solder wick or a vacuum desoldering device can be used for this step.

Place the hollow drill over the protruding pin at hole 61 and drill through the motherboard from the solder side until the bit just breaks through the board on the connector side. If you are careful, and enough solder was removed from around the pin before starting, a clean hole should result with the ground trace cut from the ground plane and the pin left isolated.

After completing the surgery on all 12 connectors, the isolated pins can be jumpered together on the solder side of the board. It is a lot of work but the pin 61 problem will not interfere with any future plans for expansion.

###



Alien Equipment

Warren Lambert, Editor

Should you buy Guns or buffers?

In this issue's Alien Equipment, Randall Staponski reviews an external print buffer that speeds up system performance by parallel processing. A Z80 central processor, generally running 8080 code, has its bitty hands full running programs like Word Star; simultaneous (background) printing while working on something else goes slowly, and home caliber printers aren't fast. While I like my cheap Epson RX-80 dot matrix printer very much, the bad news is that it is slow. Even worse news is the price of the big guns, the fast printers, which can exceed \$2000. Adding insult to injury is the fact that numbers like "100 characters per second" aren't accurate estimates of real throughput.

Bread & butter estimates

To estimate real throughput you must test a printer under working conditions. I did a naive benchmark pitting a fast 100 char/sec Epson RX-80 against a slow NEC Spinwriter rated at a mere 55 char/sec; both machines printed the same (1024 byte, half page) memo under Word Star. The Epson should only take 55% of the time, right? Wanna buy a bridge?

The smoking pistol

The NEC took 26 seconds while the Epson took 43 seconds. That's about 39 char/sec for the 55 char/sec printer and 24 char/sec for the 100 char/sec printer. Thank you for the swell bridge, but no thanks.

While the Epson might be faster for letters only one character long, the NEC has a buffer, a much heavier paper feed, and (most important) fast "logic seeking" that enables it to ignore spaces and shoot right down to the spot where the next printing occurs. The NEC takes blanks at 120 char/sec; in practical tests, there will be many blank areas on the page (assuming you're not Scrooge). So if you're printing half page memos, the 55 char/sec daisy

wheel is much faster than the 100 char/sec dot matrix printer.

When I first looked at the RX-80, I thought it would provide fast copy, e.g. for compiler listings, but actually it is slower than the Spinwriter. Its low price, cheap ribbons, and graphical capabilities make it worth while, but not its speed. I love to use it with a program called "Fancy Font," to write in Old English, 40 point Roman, and other fun fonts. Maybe I'll open a diploma mill someday.

Another edge the Spinwriter had was its 256 character buffer, not a huge buffer, but large enough for a few lines. The RX-80 only buffers one line at a time; a single line buffer lets an unbuffered computer such as the Horizon parallel sit idle, doing nothing but waiting for the RX-80's "Ready" signal. The printer can't gun its way through blank lines. Pricey new printers have buffers, but if your old faithful printer still shoots straight, Randall Staponski will show you how you can add a buffer to it--even if you can't pronounce "Ciarcia."

Encouraged by Randall's review, I ordered one, but (miracles of government purchasing!), I received a very similar Quadram parallel-parallel Microfazer instead. With some homemade plugs, the same parallel printer can be plugged into my North Star, or into a PC-clone, or the real grey flannel PC at work, all using the same standard IBM-PC (Parallel 36 to DB25) cable. (I made a N* parallel to DB25 adapter, so the Epson thinks my N* is an IBM PC. Sorry Charlie!). I added 48K of bargain basement dynamic rams (six 4164's of three different brands at \$5 each) for a total 64K buffer which accepts a 1K Word Star file in a few seconds. The Quadram parallel buffer works fine, and I'd like to thank Randall for his careful review, as without it I wouldn't have dared to bolt such strange Alien Equipment on my poor little RX-80. With the buffer, a 30 line Compass column took 12 seconds to leave the computer (46 seconds to finish printing); the Spinwriter took 31 seconds to leave the computer (33 seconds to finish).

R E V I E W

Product: 8K Buffer and Serial
Interface Board for
Epson Printers
Vendor: Practical Peripherals,
Inc.
31245 La Baya Drive
Westlake Village, CA

91362

Reviewer: Randall Staponski,
3003 S. Joplin Place
Tulsa, OK 74114

This review is intended mainly for those readers who, like myself, are uncomfortable with making major hardware modifications but would still like to become more familiar with their hardware as well as save a few bucks. After purchasing my Advantage I chose the Epson MX100 printer in order to utilize the graphics screen dump program which came with the operating system. While shopping for the printer it quickly became apparent that most dealers did not stock the serial interface and in most cases would not discount the price even though the printers were discounted.

A request to Practical Peripherals brought a reasonably written set of brochures which indicated the serial interface is compatible with the Epson MX, FX, and RX Series printers. Versions of the board are available with a buffer size of 8K, 16K, and 32K. The 16K and 32K versions are expandable to 64K. Since most of my printing consists of graphics and relatively short letters to my congressmen I chose the nonexpandable 8K version.

Physical installation is simple since the serial board simply plugs into the space where the Epson board is normally installed. The Epson manual provides very detailed instructions in opening the printer case in order to set the internal printer characteristic dip (dual inline pin) switches. Switch sw2-8 must be set to ON for the printer to work with the serial board. The remaining switches may be set as the user desires. The serial board plugs in directly above the circuit board which houses the printer characteristic switches. It is fastened with 4 Phillips screws. In order to change the printer switches at a later date the serial board must be

removed to provide access. Power for the serial board as well as a physical enclosure are provided by the printer. The only tool needed to open the printer case and install the serial board is a small Phillips screwdriver.

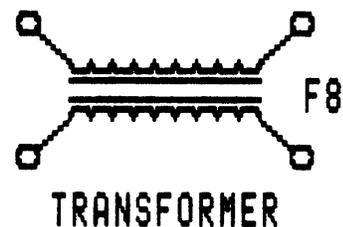
All elements of bliss ended when it came time to set the switches on the serial board itself. This set of switches configures the serial board to receive data from the host computer. Instructions for the serial board gave a good description of each switch. However, the Advantage user manual was silent on the requirements for the serial interface. Despair changed to hope after rereading North Star Answers by Robert Cowart in Compass, Vol. III, no. 2. From this the switches were set for hardware handshaking with normal busy. Word length is 8 bit, 2 stop bits, no parity, and baud rate of 9600.

The serial board uses Berg jumpers to set each switch. This jumper consists of a wire encased in a plastic housing. To turn a switch, on the Berg jumper is plugged into the appropriate socket. The jumper is removed to disable a function.

A word of caution. As indicated above, the company's literature says its serial boards are compatible with the MX, FX, and RX series printers. However, actual installation instructions made no mention of compatibility with the FX series. In addition RX80 and RX100 users were asked to call the Customer Service Department for special instructions. For most users this would be a toll call. Also, it was necessary to replace the screws on the cable connector with longer ones with a length of about 3/4 inches.

The list price for the 8K version is \$159, but with discounting the price was \$115. Graphics and text have been printed for about 3 months with no mishaps. I even received one reply from my Senator.

#



Commentary

By Steve Noll

Congratulations! Volume IV, no. 1 is the best format yet! Now here are some questions and comments for readers of Compass, Vol. IV, no. 3:

Book on NS BASIC?

An article in a 1981 issue of Popular Electronics mentioned a book titled A User's Guide to North Star BASIC by Robert R. Rogers. Does anyone know if this is still available?

Computer Insurance

One day I got to thinking of the investment I had in computer hardware and software. With an Advantage, a Horizon, hard disk, a half dozen floppy disk drives, printer, modem, software, etc., there was over \$10K. Apparently, most "Homeowners" insurance doesn't cover this stuff if it is used in a business, as some of mine is.

I decided to check out the "Safeware" insurance that I had seen advertised several times in BYTE. Their flyer was most impressive. It said they offer full replacement coverage of both hardware and purchased software, regardless of use, whether personal or business. It covers theft, fire, accidental damage, water damage, earthquake, even power surges! The deductible is only \$50. The flyer listed a dozen rates, from \$35 for up to \$2000 to \$215 for up to \$36,000. I checked my local independent insurance agent. His rates were almost three times as much as Safeware's.

For some reason, Safeware would not send me a void policy to examine. I sent in my money anyway, as they have a 10 day examination period. The policy that came was actually written by Fireman's Fund. I had my Farmers Insurance agent read it over, and he said, "Go for it!"

What are the strings? It doesn't cover data or media that can't be replaced with other of like kind currently for sale. It seems then to cover, for example, your copy of WordStar. It doesn't cover

property rented to others or held for sale. It doesn't cover wear and tear, faulty design, dishonest acts, programming errors (rats!), war, nuclear snafu's, theft from a vehicle without signs of forced entry, etc. In other words, no big surprises.

Interestingly, they never asked me for a detailed list of what I was insuring. That's nice. I have not yet had to make a claim, so I can't attest to their services in that department. The phone number of Safeware is 1-800-848-3469.

SCAN & REN

A reader recently inquired about the availability of the SCAN & REN programs I mentioned in Compass, Vol. IV, No. 1. SCAN, RENUMBER, CRUNCH, UNCRUNCH, and PSIZE comprise a package of utilities for North Star BASIC sold by E.T. Software.

SCAN will search for most any character, symbol, word, or group of words in a NS BASIC program. Any line containing the searched-for item will be listed in its entirety. For example, typing:

```
SCAN,GOSUB 1000
```

may result in

```
510 IF A>Q1 THEN GOSUB 1000 ELSE
      RETURN
705 IF Q$="THX 1138" THEN GOSUB 1000
```

But the following line would not be listed because of the missing space:

```
110 !A$/GOSUB1000
```

I find SCAN invaluable in writing large BASIC programs, especially for finding all occurrences of a given variable.

RENUMBER is a worthwhile expansion on the North Star REN command. One of the several formats of REN is:

```
REN (<Starting line #>,<Ending line #>
<New starting line #>,<Incr.>
```

Whereas running

```
REN(100,200)1000,10
```

would change:

```
10 REM ** TEST
50 GOSUB 100 STOP
100 !"Hello"
110 OUT 34,255
200 IF INP(34) <> 0 THEN 200
500 RETURN
```

to:

```
10 REM ** TEST
50 GOSUB 1000 STOP
500 RETURN
1000 !"Hello"
1010 OUT 34,255
1020 IF INP(34) <> 0 THEN 200
```

REN allows one to move a line or a large block of lines into or out of the middle of a program.

CRUNCH removes spaces and REMs to compress a program. The CRUNCHED program can be SAVED and REN, while operating faster and consuming less space than before. UNCRUNCH allows editing of a CRUNCHED program.

PSIZE lists the size of a program in file blocks, bytes, number of lines, and last line number used.

The programs load themselves automatically into your copy of BASIC. They are summoned just as any other command is. These utilities have not been modified to run under the new 2.1.1 DQ DOS/BASIC.

The SCAN & REN package is available for \$65 postpaid plus 6% tax for California from:

E.T. Software Services
1474 La Jolla Place
Carpentaria, CA 93013
(805) 684-8259

(Specify Advantage or Horizon!)

Computer Applications

I think it would be interesting if readers of COMPASS would share with us some of the applications that they use their computers for. I have a couple that may be of interest.

Some of the advantages of the S-100 bus are that it is versatile as well as well defined. The result is that it is easy to homebrew accessories for a S-100 computer. With the help of books such as Osborne McGraw-Hill's Interfacing to S-100/IEEE 696 Microcomputers I have built several parallel ports, an 8 channel A to D converter, a pair of D to A converters, and a real-time clock/calendar for my Horizon.

I recently had the opportunity to put some of this equipment to work on a real-world problem. A fairly sophisticated burglar alarm control panel I had installed was acting up. It would arm itself and then give false alarms. The manufacturer could not find anything wrong when it was sent back for repair. Apparently they couldn't reproduce the problem. I needed proof that there was a malfunction in the unit, so this is where I put the computer to work.

I connected 4 bits of a parallel port to the alarm control panel "on the bench." A quick BASIC program was written which would look for any change in the 4 lines from the panel. A change would result in the status of all 4 lines being printed on a printer along with the time and date:

```
STATUS: Bell OFF
Prealarm OFF
Alarm ARMED
Sensor warning OFF
AT 09:06:38 AM
04/01/84
Sunday
```

Thus, the needed proof (70 false alarms in 17 hours) was obtained with the computer doing all of the boring work.

Another application for my Advantage is drawing machine control diagrams (see samples). Control diagrams are schematics for electrical equipment containing such things as pumps, conveyors, heaters, temperature controls, etc. I design and

build the electrical systems for machinery manufactured by one of my clients.

Although the company has a couple of experienced draftsmen, they are mechanical, not electrical or electronic, draftsmen (a big difference). I used to have to make a very accurate sketch of a new design, give it to a draftsman to be redrawn on vellum, and then later check it for mistakes. It sometimes seemed that I might just as well have done the whole process myself.

A perfect application for a good graphics computer like the Advantage! I wrote a program in North Star Graphics BASIC to do the drawing. The program has over two dozen schematic symbols built in (fuses, resistors, float switches, pilot lights, etc.) It also draws rectangles, circles, solid and dotted lines, text, and other things needed for a

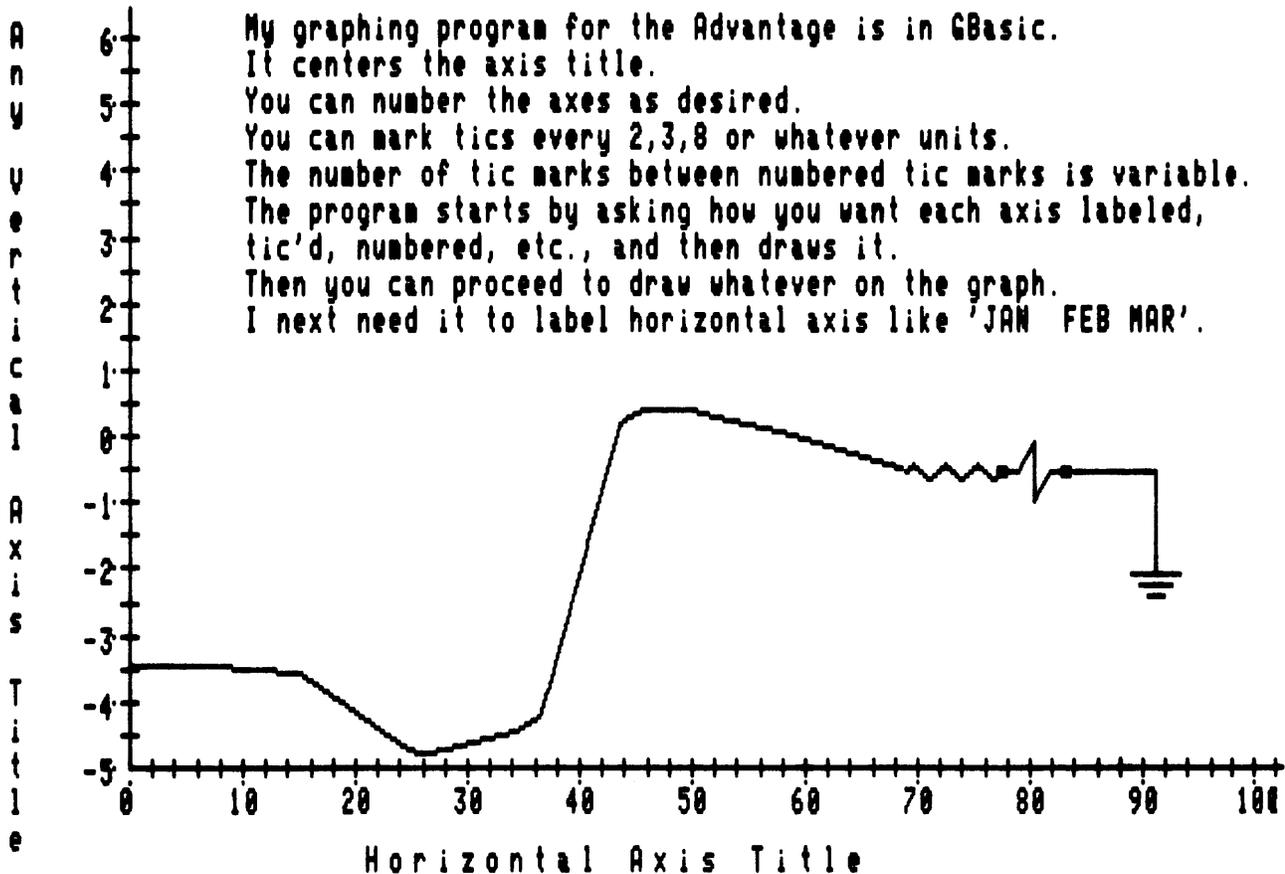
complete schematic. The program creates graphic files, maintains a directory, allows editing of existing files, and prints the schematics on a NEC 8023 printer.

To produce a new schematic, I do my designing on the Advantage, giving the draftsman the printout. He merely takes it to a Reproduction company where a xerography process yields a vellum from which the final line prints are made. Now I have a library of schematics on disk that I can edit to develop new designs with much less work.

Final Note:

I hope to have some news soon on a new PROM for the Advantage which speeds up the slow keyboard.

#



Vendors Column

In this column we reprint vendors' descriptions of their own products. Of course, INSUA makes no claims about the quality or merchantability of any commercial software, hardware, or services. **Caveat emptor!**

INSUA:

I think that your readers would benefit from knowing about my Company, Fischer Computer Systems. We specialize in the repair and upgrading of North Star Hardware. We offer what we call a System Tune which is the complete repair including modifications recommended by North Star and a ninety day warranty on the whole system for a flat fee. We also can do any level of upgrading which is necessary. Our prices and turn around time are very good. We have shipping containers which we will send out to allow the safe shipping of a micro-computer from anywhere.

I used to work for North Star Computers, but since 1980 I have been in business for myself working on North Star Hardware. Fischer Computer Systems has been doing production board repair for North Star Computers for four years and as a result has the best resources for repairing North Star Hardware. We are not too worried about strange boards in your system. If Documentation is available for the boards we can probably repair and test them; if not we will make sure the system works and warrantee everything except the strange boards. We will also work on any system which contains a North Star floppy disk controller with North Star DOS.

We have several things related to TurboDOS which should be of interest. One is the software and hardware for a 128K slave Z-80 board. This allows one to have a 62.7K TPA compared to North Star's 49K to 53K with the UP-8. We also have a single user version of TurboDOS running with two bank-switched H-RAMS in the Horizon. We do a small hardware modification which allows us to place the disk controller in bank 0 and have a complete uninterrupted 64K in bank 1. With this arrangement we have a 62.7K TPA. A small amount of the available RAM

in bank 1 is used for bank-switching overhead and that is why there is not a full 64K TPA. Using this arrangement there is plenty of RAM for future operating system expansions without taking RAM away from the user. With TurboDOS the extra RAM in bank 0 is used for disk buffering, which helps to speed up system operation.

Fischer Computer Systems also has several software products which should be of interest. A patch for DOS to allow the use of SemiDisk Systems' 512K disk simulator. A Modem7 compatible program which runs under DOS. A full screen text editor written completely in North Star BASIC. An easily modified data entry program which allows definition of fields and prompts for building data bases, etc. A disk simulator which uses the 8088 board and RAM in an Advantage under CP/M. (This should be of special interest since North Star has drastically reduced the prices on that product.) X-on x-off or ETX-ACK software hand-shaking routines for CP/M on either the Horizon or Advantage.

We also have several hardware products. An eight inch floppy disk interface and a composite video interface for the Advantage. For the Horizon (S-100 buss) we have an optical encoder counter interface. Also a day/date clock for the Horizon which does not use an S-100 slot.

For pricing and information on what Fischer Computer Systems can do for you, please contact Randy Fischer at: Fischer Computer Systems, 445 Bay Street, Angwin, CA 94508, (707) 965-2414.



GRAF LISTING

```

PROGRAM GRAF;
CONST {these values apply to NorthStar Advantage}
      CH=8;      {Character height in number of dots.}
      CW=7;      {Character width in number of dots.}
      XDOTS=639; {Maximum number of x dots.}
      YDOTS=199; {Maximum number of y dots.}

TYPE
  STRG=STRING[80];
  SYMCD=PACKED ARRAY [1..7] OF BYTE;
  SYMARRAY=ARRAY [0..5] OF SYMCD;

VAR B,C,D:CHAR;
    SYM:SYMARRAY;
    S:STRG;
    I,X,Y:INTEGER;

PROCEDURE VIEWPORT;
CONST GRAF=12;
VAR
  VERTICES:ARRAY [1..4] OF INTEGER;
BEGIN
  VERTICES [1]:=0;
  VERTICES [2]:=0;
  VERTICES [3]:=XDOTS;
  VERTICES [4]:=YDOTS;
  INLINE ($3E / $02 /
    $0E / $81 / {CHANGE $81 TO $80 IF FRAME IS NOT WANTED}
    $21 / VERTICES /
    $CD / GRAF );
END;

PROCEDURE DOT (X,Y: INTEGER);
CONST GRAF=12;
VAR VERTICES:ARRAY [1..2] OF INTEGER;
BEGIN
  VERTICES [1]:=X;
  VERTICES [2]:=YDOTS-Y;
  INLINE ($3E / $01 /
    $06 / $01 /
    $0E / $01 /
    $21 / VERTICES /
    $CD / GRAF );
END;

PROCEDURE LINE (X,Y: INTEGER);
CONST GRAF=12;
BEGIN
  Y:=YDOTS-Y;
  INLINE ($3E / $06 /
    $0E / $06 /
    $2A / $01 /
    $0E / $01 /
    $21 / VERTICES /
    $CD / GRAF );
END;

PROCEDURE SYMBOL (X,Y,I: INTEGER);
PROCEDURE SYMPT (S: SYMCD; X,Y: INTEGER);
CONST GRAF=12;
BEGIN
  Y:=Y-4;
  X:=X-4;
  INLINE ($3E / $05 /
    $06 / $20 /
    $0E / $82 /
    $2A / Y /
    $5D /
    $2A / X /
    $DD / $21 / S /
    $CD / GRAF );
END;

PROCEDURE WCHAR (X,Y: INTEGER; C: CHAR);
CONST GRAF=12;
BEGIN
  Y:=YDOTS-Y;
  Y:=Y-7;
  INLINE ($3E / $05 /
    $2A / C /
    $45 /
    $0E / $02 /
    $2A / Y /
    $5D /
    $2A / X /
    $CD / GRAF );
END;

PROCEDURE WSTRING (X,Y: INTEGER; ENTRY: STRG);
VAR I,K: INTEGER;
BEGIN
  K:=LENGTH(ENTRY);
  FOR I:=1 TO K DO
    BEGIN
      WCHAR (X,Y,ENTRY [I]);
      X:=X+CW;
    END;
  END;
END;

```

```

PROCEDURE INITSYM;
VAR I,J:INTEGER;
BEGIN
FOR I:=0 TO 5 DO
FOR J:=1 TO 7 DO SYM[I,J]:=0;
SYM[0,4]:= 8;
SYM[1,3]:= 8;
SYM[1,4]:=28;
SYM[1,5]:= 8;
SYM[2,3]:=20;
SYM[2,4]:= 8;
SYM[2,5]:=20;
SYM[3,3]:=28;
SYM[3,4]:=28;
SYM[3,5]:=28;
SYM[4,3]:= 8;
SYM[4,4]:=20;
SYM[4,5]:= 8;
SYM[5,3]:=28;
SYM[5,4]:=20;
SYM[5,5]:=28;
END;

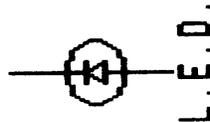
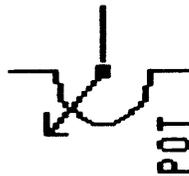
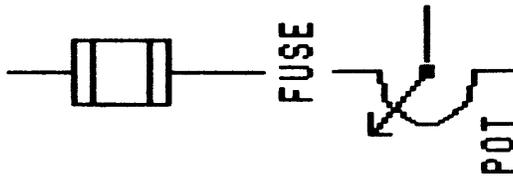
BEGIN
INITSYM;
WRITE(CHR(18)); {ENTER HYBRID MODE}
VIEWPORT;
REPEAT
WRITE(CHR(7));
WRITELN('L=LINE, D=DOT, C=CHAR, S=SYM, G=STRING, 0=STOP');
READ(D);
WRITELN;
CASE D OF
'L': BEGIN
WRITELN('ENTER X, Y');
READLN(X,Y);
LINE(X,Y);
END;
'D': BEGIN
WRITELN('ENTER X, Y');
READLN(X,Y);
DOT(X,Y);
END;
'C': BEGIN
WRITELN('ENTER X, Y');
READLN(X,Y);
WRITELN('ENTER CHAR');
READLN(B);
WCHAR(X,Y,B);
END;
'G': BEGIN
WRITELN('ENTER X, Y');
READLN(X,Y);
WRITELN('ENTER STRING');

```

```

READLN(S);
WSTRING(X,Y,S);
END;
'S': BEGIN
WRITELN('ENTER X, Y, SYM');
READLN(X, Y, I);
SYMBOL(X,Y,I);
END
END;
UNTIL D = '0';
WRITE(CHR(19)); {RETURN TO TEXT MODE}
END.

```



Letter to the Editor

By Roger Barnhill

Being a new member to INSUA this will be my first attempt at contributing to The Compass. However it will not be the last.

Let me start by saying I have ordered all of the past issues of The Compass. I have read all of them through once and am now starting over. Any of you other new members who can rake up the bucks would find your money well spent.

To The Compass editor, let me say **WELL DONE** in spite of some of the remarks I have read. I will say that I like the later format much better than the early versions. For those out there that don't care for some of the articles (Wordstar/Epson) that have appeared from time to time all I can say is, you are free to contribute anything that you think will be of more interest.

I too have some well chosen words to say about this !%?c#i* @ Word Star. I think that as long as these machines have been around there has to be something better; however it is still better than quill and ink.

At this time let me put in my vote for **not** publishing or selling the mailing list of INSUA members. You are very misinformed if you think the sellers of TRS-80 disk drive oil and all the other gimicks out there won't be at your door. If the vendors of North Star related products want to contact me I would as soon they do it with paid advertising in The Compass.

This system is a HORIZON with 2-Q Tandon 100-2 drives, upgraded to 18-Mbyte hard disk, running under TSS/C. The drives turn on and off intermittently. Something like 10 seconds on then 30 seconds off. They will do this without any input to the keyboard, that is, all by themselves. Some of the other users may be having this problem and not know about it. The little red eye does not come on. The only way you will know about it is if you operate with the cover off and watch it. (The manual says not to do this.)

I asked Frank Khozad about this problem. He said it may be incompatible

disk drives. We swapped drives, and nothing changed. I personally think it is incompatible chips on the controller board. Probably some chips from Nicaragua installed too close to some chips from El Salvador.

Now it is common knowledge that floppy disks will experience physical wear. Let's suppose you boot up the system early in the morning and leave the system disk in drive #1 all day, then cut the power late at night with the disk still in the drive. Early the next morning when you turn the power on with that disk still in the drive and the system fails to boot, are you going to claim that turning on and off the power zapped the disk? or was it because the disk got eight hours of physical wear the day before? We cannot say that we have ever lost a disk from powering up or down with a disk in the drives.

Help Wanted!

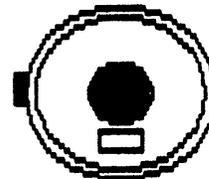
And now a request for advice. We have a North Star HORIZON with 2-Q drives and an 18-Mb hard disk upgrade running under TSS/C using HBASIC applications programs.

We run several programs that use the INCHAR\$(0) statement inside a DEF FN statement.

A problem arises when user #2 is running one of these programs and user #1 types a C at the keyboard. When this occurs the program in progress at user #2 will abort.

Northstar is aware of this bug but they seem to wish to ignore it. I feel that the bug is in the operating system. Has anyone else experienced this problem? Does anyone have a solution?

#



**COMPASS ADVERTISING
RATES**

The Board of Directors of INSUA has agreed to solicit advertising for Compass. INSUA is particularly interested in advertising for North Star compatible hardware, software, publications, etc.

Rates, until further notice, are as follows:

Full page: \$225.00
Half page: 125.00
1/4 page: 90.00

Current circulation of Compass is about 1,500 copies six times per year. The current average print-run is 2,500 copies. All correspondence should be directed to INSUA, P.O. BOX 2910, FAIRFIELD, CA 94533

Computer Pollution Solution

EASY-CARE FAN FILTER KIT
NOW AVAILABLE FOR YOUR
NORTHSTAR HORIZON®

or other hardware using 4-1/2" box-type fan

- * Fan Filter Kit eliminates dust and dirt from computer and disk drives. Pressurizes cabinet with filtered air to keep dirt out instead of sucking dirt in.
- * Water-washable Scottfoam® Filter Element easily serviced without tools. Spare elements available in 30 pore-per-inch and 45 pore-per-inch density.
- * Hole Plug Kit for Horizon® increases cooling efficiency by forcing air through card cage.
- * Quiet, high-performance ROTRON WHISPER® XL fan available to upgrade Horizon® fan.
- * Not expensive: Basic kit \$20, Extra element \$5, Horizon® hole plug kit \$7, Fan \$30, Fan cord \$2

Call or write for free reprint of COMPASS (Volume IV, No. 1) article and ordering information:

the Sound Steward

1606 WEST SOUTHGATE AVENUE · FULLERTON, CALIFORNIA 92633 · PHONE (714) 871-6636

Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203



TATE 276H
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

The Compass

International NorthStar Users Association

Volume IV No. 4



Notes from the Editor

Disks late?

INSUA owes an explanation to members for the enormous delays in delivering various disks, whether part of free disk offers or orders from the disk library. For the free disk offer, the company hired to produce the disks ordered the disks from "back East." The non-arrival of the disks was not discovered for some time either by the company or by INSUA. Finally tracers were sent out, the disks could not be found, new disks were ordered, and so forth and so forth. We think that the promised disks will be mailed out before this issue gets into the post, but we have learned not to count our chickens before they are hatched. In a way the problem is a version of an increasing difficulty noticed by one of the commentators in this issue, the difficulty of finding hard-sectored disks. We do not want to exaggerate this problem, however, since in this case the problem had to do with delivery and not with manufacture.

No documentation?

Members who receive disks from INSUA sometimes write to complain about the lack of documentation. INSUA's policy is to provide documentation **on disk** rather than on paper wherever possible. If you get a disk from INSUA, inspect the files on the disk with **LI** (in DOS) or **dir** in CP/M. Almost always you will discover a document file such as READ-ME in DOS, or xxxx.DOC in CP/M. Use your normal text-reading facility in DOS, BASIC, or CP/M to inspect the documentation. Some documentation is so simple that it can simply be read and comprehended on the screen. Other documentation is more complex, and should be printed on paper for more concentrated reading. INSUA feels it does enough publishing with Compass--disk documentation is neat and easy for all concerned.

R.I.P.

The INSUA Board of Directors laments the passing of Microsystems and Microcomputing, two magazines which rendered a great service to North Star owners. We understand that Dr. Dobb's is still being published, and may pick up some of the slack. Things they are a-changin', and INSUA recognizes this. The Board of Directors invites members' opinions on the new directions INSUA and Compass might take. Should we retrench and reassert our continuing allegiance to North Star DOS when North Star itself is abandoning its own DOS and BASIC? Should we attempt to serve purchasers of the North Star Dimension? Should we become a CP/M magazine? and S-100 magazine (at the expense of Advantage owners)? Should we try to be all things to all people?

Wanted

Whatever the future holds, the present holds a need for articles on the wide spectrum of North Star topics, particularly since we are still aiming for a total of six issues during the current membership year, plus a special issue. Articles are invited on S-100 boards, routine maintenance, dBASE programs, new hardware, methods for transferring files between alien systems, programs for solving personal and business problems. Reviews of any program or device not yet reviewed in Compass are very much in order! Please submit articles on a single-sided "North Star" disk, and include a paper printout as well.

The Compass

The Compass is published every two months by INSUA, the Interational North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, Compass, P.O. Box 2910, Fairfield, CA 94533.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

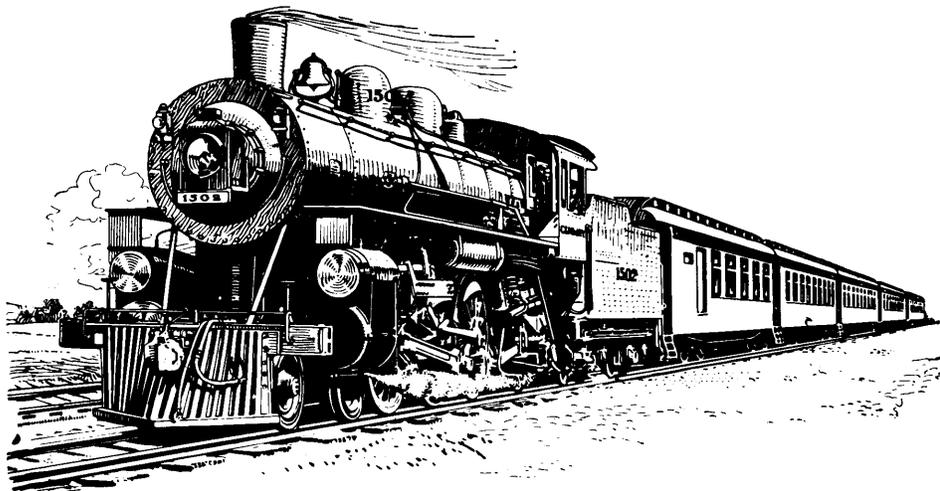
DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

Contents

- 2 Bob Beaver **dBASE a la Menu**
A general menu for any dBASE master program
- 4 Joe Maguire **Run Your Horizon at 6MHz**
Upgrade your Z-80 chip to speed up your system
- 6 Warren Saunders **Commentary**
The Phantom Signal and Fan Filters for the Horizon
- 7 Daniel D. Stuhlman **Horizons and Apples**
Bench-test comparisons for computers and programs
- 12 **Bugs Bugs Bugs**
Hugh Power reports a bug in 5.2 DOS
- 13 Alan H. Nelson **Look Before You Leap**
PC-DOS compared to CP/M
- 15 **Advantage Answers**
Function keys, Windows, 8/16, Documentation
- 16 Terry Hough **Commentary**
HDOS File Manager and North Star BASIC
- 17 Joe Maguire **Disk Controller PROMS**
Decoding North Star's Disk Controller PROMS
- 19 Saul Levy **A File-Size-Changing Program**
FILESIZE Program (with listing)
- 23 **Zips & Names**
Zipcode listing of current INSUA members
- 28 Charles Prohaskas **Disk Simulator Patches**
Patches offered for the LS-100 Disk Simulator
- 29 **Vendors Column**
Micro Support, TIS-APL, Eye-Care CR-39, TurboDOS
- 31 **Letters to the Editor**



dBASE a la Menu

by Bob Beaver

Perhaps you've been looking for a bow to tie around those dBASE command files you've been so carefully collecting. If so, read on and discover dMENU, a general purpose menu program for dBASE.

Menus are often used in programs to provide a user friendly method of selecting program options. For example, instead of the cryptic "READY" or the dreaded "*" prompt, the user is greeted with something like

-OPTIONS-

1. Append Transactions
2. Edit Selected Transactions
3. List Transactions (Journal)
4. Recap an Account
5. Reconcile an Account
9. Exit to DBASE II

-Select Option (1-9):

and even if you don't know what "Reconcile an Account" means, at least you can press 5 and watch the fireworks.

Also of note is that a menu is helpful to the programmer, since it provides a framework for program development and a 'to do' list which can be completed in sections. The menu can also be modified or expanded as needed.

Should all programs have menus ? Should books have tables of contents ? Does a bear...., on second thought, I leave that question for future lively debate and press on with dMENU.

The program listing which follows provides an example which can be easily modified for your dBASE .CMD files using WORDSTAR or other CP/M word-processor.

The 'envelope' for this program consists of two statements, a beginning (DO WHILE 'A'='A') a condition which is always true, and the ending (ENDDO). The program executes the statements between these two lines until an escape key is pressed or a RETURN statement is executed from within the program (see statements for option 9 in the listing).

A few simple housekeeping commands follow the DO WHILE statement (CLEAR

to ERASE). All of these statements are optional, but useful.

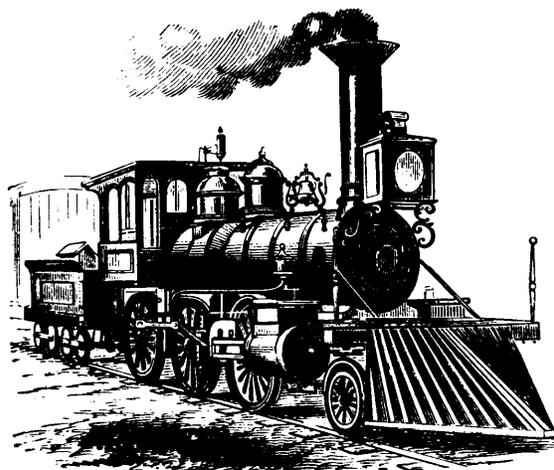
The menu display and selection prompt follow (? to ACCEPT). These can be revised as desired.

The remaining program lines execute dBASE command files. Substitute your own .CMD file names for the ones shown in the DO statements, and provide a drive specification (e.g. DO C:GLAPND) if the command file is on a drive other than the default drive.

When the .CMD file completes, it will return to the dMENU program automatically, permitting another option to be selected.

As a parting note, the simplicity of this program and others written in the dBASE command language still impresses and delights me after several years of use (I bought an 8K IMSAI computer in '74 and for a while would toggle-in bootstrap programs using the front panel...Ah, for the good ol' days). In spite of possible complaints of the slowness of dBASEII, it has provided a powerful tool to the world of personal computing.

Enjoy dMenu!



```

* -----
*           dMENU.COMD - Menu Command File for dBASE

do while 'A'='A'
clear
set default to b:
set talk off
erase
?
? '           *** General Ledger Master Menu ***'
? '                                     ',date()
?
? '           -OPTIONS-'
?
? '           1. Append Transactions'
? '           2. Edit Selected Transactions'
? '           3. List Transactions (Journal)'
? '           4. Recap an Account'
? '           5. Reconcile an Account'
? '           9. Exit to DBASE II'
?
accept ' -Select Option ' to ms

if ms='9'
?
? 'Returning to DBASE II.'
return
endif
    if ms='1'
    do GLAPND
    loop
    endif
        if ms='2'
        do GLEDIT
        loop
        endif
            if ms='3'
            do GLLST
            loop
            endif
                if ms='4'
                do GLRECAP
                loop
                endif
                    if ms='5'
                    do GLRECON
                    loop
                    endif
?? ' * Option Error '
enddo
* End of dMenu
* -----

```

Run Your Horizon at 6MHz

By Joe Maguire

With all the hullabaloo over the so-called superiority of new computer models these days, I decided to see what I could do to improve my old, reliable Horizon. One possibility was the speed of the Z80 CPU.

When the Horizon was first introduced, a 4MHz Z80 was the fastest available. Nowadays, six and even 8MHz versions are available. Could my Horizon handle that? I decided to find out.

CPU speed is determined by two things: the system clock and chip architecture. In the Horizon, the system clock is derived from an 8MHz crystal oscillator on the CPU board. The crystal frequency is twice the clock, and CPU speed, of 4MHz.

The CPU manufacturer designs the chip for some maximum speed by using special fabrication techniques. At present, there are four versions of the Z80. They are:

Z80	2MHz
Z80A	4MHz
Z80B	6MHz
Z80H	8MHz

The Horizon comes equipped with a Z80A. I decided to try a Z80B at 6MHz.

The System Clock

The CPU clock is also the master system clock. That is, various signals are sent from the CPU board, over the S-100 bus lines, to other boards and the motherboard. The motherboard uses a 2MHz signal, from pin 49, to clock the serial port USARTs to the selected baud rate. It is also gated to the interrupt circuitry to set the interrupt rate, if used.

If you are using a NorthStar dynamic RAM board, the 4MHz phase 1 and 2 clock signals from the CPU board are used to control the refresh rate. More about this later.

In a basic configured Horizon, (i.e. CPU board, memory board, disk controller) these are the only users of the system clock. The disk controller has its own, onboard clock.

Changing the crystal frequency on the CPU board will change the system clock rate. For users which require a specific frequency, i.e. the USARTs, we must provide an alternate clock source.

The Changes

Two changes are required on the CPU board. First is to replace the Z80A with a Z80B, an easy job. Second is to replace the 8MHz crystal with one for 12MHz. This requires a little soldering. Don't overheat a crystal; they can be damaged that way. Just apply the tip of a soldering pencil long enough to remove the leads from the board and the ground strap from the case. Install the new one the same way. I found that I did not have to change any other components in the oscillator circuit for it to work at the higher frequency.

This crystal change will raise the system clock rate to 3MHz and the phase 1 and 2 signals to 6MHz.

To get the USARTs back in sync a 2MHz signal must be found and routed to S-100 bus pin 49. This can best be taken from the controller board. A 2MHz signal is available from several points on the board. A convenient connection is pin 4 of IC 12C. If you jumper this to pin 49 on the controller board, you must cut the trace to pin 49 on the CPU board.

If you are hesitant about cutting traces you may, instead, prefer the method described in the hardware note by Bob Stek in Compass Vol. 1, No. 2, p. 17. (See also correction in Vol. 1, No. 3, p. 17)

In my own system, I just bent pin 11 of IC 7C, on the motherboard, out of the socket and jumpered it with wirewrap wire to pin 4 of IC 12C on the controller board. If you have made the 19.2K Baud modification to your system, as described in the above reference, and require a 4MHz signal to the USARTs, it is available at pin 10 of IC 15B on the controller board. Put the controller board in the rearmost slot of the motherboard for this option.

If any other boards in your system

require a 2MHz signal on bus pin 49, i.e. cassette interface, modem board, etc., you should use the jumper-to-pin-49 option.

This still leaves us with the problem of the phase 1 and 2 signals. There is no way to recover them at 4MHz without adding additional hardware. In the basic configured Horizon, these are used for refreshing NorthStar's 16K and 32K dynamic memory boards. The HRAM doesn't use phase 1 and 2 but takes its refresh signal from bus pin 66 instead.

I was using a RAM 32-A1 in my system (modified for 64K), with 150ns memory chips, but I found it would not work with the CPU running at above 5MHz. The problem is apparently in the refresh circuitry and not the with the chip speed. (I tried adding a wait state) Fortunately, I had a 64K static board on hand (from Digital Research of Texas) which worked fine so I just used that instead.

One problem surfaced while testing the system with the 6MHz CPU speed. The DD disk controller. I could only get satisfactory disk operation by adding the wait state jumper on the CPU board. (near IC 7G) This reduces the speed gain somewhat but there is still a noticeable improvement.

Faster! Faster!

I was so elated with my new computing power that I became reckless. I knew that most CPU manufacturers are conservative about the ratings of their chips and I suspected that 6MHz was not the upper limit for the Z80B. In rummaging around in my junkbox I spied a 14MHz crystal. I was sure that the CPU could tolerate a clock speed of 7MHz but what about the memory and the disk controller? Oh well, fools rush in...etc.

I exchanged the 14MHz crystal for the 12MHz, plugged in the CPU board and turned on the power. It worked! There was the signon message and the prompt sign. I tried all the disk commands, loaded Basic and ran a program, exited Basic, ran a disk test--in short, every function worked perfectly!

Now I decided to throw all caution to the winds. I frantically searched my junkbox for a 16MHz crystal. No luck...but wait! There's an 18MHz crystal. No, it couldn't possibly...but what the heck. With trembling hands I soldered it in. I

plugged in the board and, with my eyes shut, turned on the power. I just couldn't bear to witness the failure. The disk whirred, the familiar clicks were heard. I opened my eyes. Good grief! There's the signon message! I typed a command line; tried deleting some characters; so far so good. How about the LI command? There are the files! I can't believe it. My ancient Horizon is running at 9MHz!!

Now I'm feeling cocky. I'm going to try my Basic program again. I type GO BASIC. The disk whirrs...CRASH! The system has locked up. I try the RESET switch...nothing. I turn off the power...wait a few moments...turn it on again. Yes! There's the signon again! I type LI. CRASH! Oh well, I didn't really expect it to work at 9MHz. Somewhat dejected I solder in the 14MHz crystal again. I guess I'll have to be satisfied with a mere 7MHz CPU speed.

I put in my disk and turn on the computer. The drive whirrs, and clicks, and cranks. No signon. Uh oh. What happened? I try another disk. OK this time. Now I understand what happened. At 9MHz I have reached the speed limit of the disk controller. When it was trying to read, it wrote instead and botched up a few bytes on the disk. I'd even had the write protect tab installed.

But the CPU and my memory board seemed to handle that speed OK. I had bought 150ns static RAM chips and, with the wait state enabled on the CPU board, I detected no memory problems.

As a matter of speculation, I think the disk controller might be able to handle an 8MHz speed. I'm going to try that as soon as I can obtain a 16MHz crystal.

Some Findings and Figures

I decided to go back and try some tests with the original Z80A CPU. Here's what I found using a simple Basic program for a speed test:

The test program:

```
10 FOR I=1 TO 10000
20 NEXT
```

Timing from RUN to READY:

Z80A - 4MHz	no wait state:	9 sec.
Z80A - 4MHz	1 wait state:	11 sec.

I tried the Z80A at 6MHz. It worked OK.

Z80A - 6MHz	1 wait state:	7 sec.
Z80A - 7MHz	(no good at this speed)	
Z80B - 6MHz	1 wait state:	7 sec.
Z80B - 7MHz	1 wait state:	6 sec.
Z80B - 8MHz	(estimated)	5 sec.

As you can see from my experimentation, all you need to get going at 6MHz is a 12MHz crystal (about \$3) and a suitable memory board. If you want to go faster, and it appears that you can, you'll need at least a Z80B (about \$10) and fast memory chips. For such a modest outlay it's certainly worth a try.

Many S-100 accessory boards have their own onboard clock generators. Increasing the CPU speed should have no effect on them. For those who remember my review of the LS-100 disk emulator in a past issue of Compass, it works fine at 7MHz in my system.

Good luck!

#

Commentary

By Warren Saunders
3417 Hickory Hills Dr.
Oakton, VA 22124

I would like to submit the following comments on your recent issue (Vol. 4, No. #1):

1. The Phantom Signal - About 3 years ago I discussed this scheme with a North Star Engineer. While it does work most of the time under most circumstances, it is not fool-proof and therefore not officially endorsed by North Star. The reason is that the source signal (Output LATCH ENABLE) being applied to the phantom line is asserted very late in the memory-read cycle. It's the signal that enables the tri-state bus driver. In other words, the RAM board at the corresponding address location goes through a full memory fetch cycle (better not have wait states enabled on this board) up to the

time that it would tri-state its data on the data bus. Then along comes the phantom signal from the controller board to abort the read process. The phantom signal just may not make it in time. Comments would be welcome from design engineers who know more than I as to how reliable this fix really is.

2. Computer Pollution Solution - I have just recently completed a similar fan reversal & filter installation modification to two Horizons. I was not ambitious enough to construct my own filter cover and select an appropriate filter material so I kept looking for a commercially available product and source of supply. After years (really) of looking I located a source. It's available from ACTIVE, PO Box 8000, Westborough, MA, 01581 (800) 343:0874. They also have stores in Boston and Seattle. Part Number is 3-90-8099, cost is \$4.40 each. A replacement filter (3-90-8106) is also available for \$2.45. These filters are made exactly for these standard 4-11/16" fans. ACTIVE'S catalog shows an exceptionally complete product line for computer and electronic parts. I've had no problems with several mail orders. Only word of caution about the installation process: the older Horizons had a single large round fan hole (about 4") with a separate finger guard grill mounted on the outside. These are easy to convert, just turn the fan around and replace the grill with the filter assembly; you may need slightly longer screws though. However, if you have one of the (newer?) Horizons with lots of small stamped-out holes in the back panel for a grille, you may be in for more work. It seems that on some of the fans the center on the air intake side sticks out a bit further than the sides. Thus when turned around the center of the fan will press against the back panel and will not rotate! You'll need to either insert standoffs or cut away the stamped-out holes making a single large hole. I elected the latter since opening up the stamped-out holes makes room for a lot more air flow. Thus the air flow loss through the filter filter? will not be significant. By the way, if you're the lazy type, plain old vinyl tape will plug up all the back panel holes just as effectively as plastic plugs and specifically-cut sheet metal.

#

Horizons and Apples

By Daniel D. Stuhlman

In 1979 when I bought my North Star Horizon, microcomputers were a novelty. Some of the early manufacturers are no longer in business. Many of the first machines were sold only in kit form. When I went to buy a computer the salesman convinced me that microcomputers were here to stay, but the future of any particular company was dubious. At the time no one knew how many micros the public would buy. To convince me to buy a Horizon he said, "Why gamble that Apple or Commodore will be around when you need them? Buy an S-100 buss machine and even if one company goes under there will still be third party vendors to supply peripherals and enhancements." Today both Apple and North Star are strong companies and many third parties make add-ons for each family of machines.

I am satisfied with my Horizon and wouldn't trade it for a whole orchard of Apples. When I was on the faculty of Kendall College I had the opportunity to compare the operation of the Horizon with the Apple II. This article compares how the operate with three languages: BASIC, COBOL, and Pascal. Without polemics the benchmarks will tell you which machine is better.

I. Execution time

Looping is an important part of most computer programs. Looping allows a series of instructions to be repeated under program control. These loops may be dependent on count or condition. Examples are: FOR I = 1 TO 10 ... NEXT I in BASIC; PERFORM I 10 TIMES in COBOL; and For I:= 1 to 10 do begin ... end; in Pascal. Notice that BASIC and COBOL allow only uppercase, while Pascal allows upper and lower case. Each language has various ways of implementing conditional loops so that the programmer can choose which is best for the situation.

Since part of the reason for using loops is to execute the contained statements exactly the same way every iteration, the speed of the machine can be tested using a loop.

The execution speed of any particular machine language instruction depends on the central processing unit (CPU) clock and the time required for a memory access. A T-state is a single clock pulse on the CPU. The Horizon uses a Z-80 microprocessor as a CPU with a 4 megahertz(MHz) clock. This means that four million clock pulses occur each second. The Apple uses a Motorola 6502 microprocessor operating at 1.023 MHz. The Z-80 CPU allows the Horizon to operate much faster because instructions do not have to wait as long between clock pulses. The number of T-states that each instruction requires can be found in assembly language books. Assembly language is beyond the scope of this article. To put perspective on the benchmarks machines other than the Apple II+ and Horizon were tested.

BASIC benchmark:

```
10 INPUT "Upper limit of the loop",K
20 FOR I = 1 TO K
30 A=A+1
40 NEXT I
50 PRINT CHR$(7),"A= ",A
60 END
```

This program asks for the upper limit of the test loop, adds 1 to A K times, rings the bell to indicate the loop is complete, and then prints A. The times were measured in seconds from the carriage return after inputting K to the hearing of the bell.

BASIC benchmark timings

iterations	computer			
	Horizon	AppleII+	IBM-PC	TRS-80 Color Comp.
1000	3.42	4.15	3.84	9
2000	6.81	8.14	7.28	17
3000	10.25	12.14	10.80	26

Since the timings were done with a stop watch and are subject to reaction time they should be considered only relative. Several trials were made and the above times represent averages. If a programmer knows these times an intentional delay can be created.

Pascal benchmark

```
Program loop;
var
  I,K : integer;
Begin
  Writeln('Upper limit of loop');
  Readln(K);
  For I := 1 to K do begin
    A:= A+1
  End; .of for loop
  Writeln(chr(7),'A= ',A)
End.
```

Pascal benchmark timings

iterations	Apple	Horizon
	UCSD Pascal	Turbo Pascal
1000	2.18	.31
2000	4.26	.26
3000	6.36	.36

Again the times are in seconds from the carriage return after inputting K to the hearing of the bell. The results using Turbo Pascal came so quickly that the times are very close to reaction time. It takes about .1 seconds to turn on and off the stop watch as fast as possible.

Since COBOL is more verbose than Pascal and BASIC only the PROCEDURE DIVISION is listed, because it contains the algorithm. The other divisions are not needed for the comparisons.

COBOL benchmark

```
PROCEDURE DIVISION.
PARAGRAPH.
  OPEN OUTPUT PRINT-REC.
  DISPLAY "Upper limit of loop".
  ACCEPT TOP.
  PERFORM PARA1 TOP TIMES.
  PERFORM WRITE1.
PARA1.
  ADD 1 TO A.
```

```

WRITE1.
  MOVE A TO PRINT-RECORD.
WRITE PRINT-RECORD.
CLOSE PRINT-REC.
STOP RUN.
END PROGRAM LOOP.

```

COBOL benchmark timings

iterations	Nevada COBOL	
	Apple	Horizon
1000	14.13	6.99
2000	26.58	13.42
3000	40.32	19.95
4000	1:20.25	39.45

The times in seconds were counted from the carriage return after inputting to the moment the printer started. It is difficult to make the system bell ring in COBOL.

The program development times vary greatly. BASIC was the easiest the write and took five minutes from the time I turned on the computer until the completion of the loops. Pascal took about ten minutes, while COBOL took over five hours.

Notice that the timings do not increase at a constant rate. This suggests that to start a loop requires a certain amount of "overhead". This overhead is independent of the number of iterations. After this overhead is ameliorized the rate comes close to being constant.

II. Floating Point Addition

Computer represent numbers with binary bits. The number of bits used to represent a number varies according to the computer, language, and the number of digits of precision. For example North Star BASIC in the standard eight digit precision stores numbers in five bytes, while Turbo Pascal stores real numbers of 11 significant digits in six bytes and integers with values from -32,768 to +32,767 in two bytes.

Binary addition is based on the following table:

0 + 0 = 0	1 + 0 = 1
0 + 1 = 1	1 + 1 = 10

Integers can easily be represented in binary. For example:

Decimal	Binary
1	0001
2	0010
3	0011
8	0100
16	1000
56	0011 1000

Problems come with arithmetic operations when the numbers contain digits on both sides of the decimal point. These floating point numbers are also called real numbers. Real numbers are usually defined in scientific notation. In scientific notation the significant digits (also called mantissa) are separate from the exponent (power of ten) part of the number. These two parts are stored separately in their own bytes. In North Star's eight digit BASIC the first four bytes of storage store one digit of the number in each nibble (half-byte). In the fifth byte bits 0-6 store the exponent. Bit 7 holds the the sign (1= negative; 0= positive). The exponent is limited to 65.

The actual scheme for adding numbers varies by computer and language. COBOL uses implied decimal points defined by PICTURE clauses. All numbers in COBOL behave like integers. There are no random numbers or built-in transcendental functions in COBOL.

North Star BASIC uses binary coded decimal (BCD) for numeric operations. In BCD each decimal digit is stored in a nibble. The following binary values are used:

```

=====
decimal digit      binary
    0              0000
    1              0001
    2              0010
    3              0011
    4              0100
    5              0101
    6              0110
    7              0111
    8              1000
    9              1001
=====

```

Binary values of 1011 to 1111 are not used.

Since the Z-80 registers can store only two BCD digits at a time arithmetic is done in a combination of register pairs and main memory. Theoretically this allows precision limited only by available memory.

Large main frames such as the IBM 370 family the programmer can choose the precision for each variable. PL/1 allows numeric types using from 1 to 32 bits. (That's a range of -4,295,177,296 to +4,295,177,295.) Most microcomputer languages do not allow a such a wide choice of numeric types. The usual choice is only real or numeric. North Star BASIC offers 8, 10, 12, and 14 digit precisions. Files written with one version can not be read by another version because as the number of digits of precision increases the number of bytes required for storage also increases.

Every COBOL and North Star's BASIC arithmetic always give accurate results when adding decimal numbers. Other microcomputer languages are not always accurate. The following benchmarks illustrate the problem.

BASIC benchmark for floating point arithmetic

```

10 INPUT "LIMIT OF THE LOOP",K
20 INPUT "GIVE ME AN ADDEND",J
30 FOR I = 1 TO K
40     B = B + J
50     NEXT I
60 A = K * J
70 PRINT A,B
80 IF A=B THEN PRINT "Equal" ELSE PRINT "Not equal"
90 END

```

From a mathematical point of view A & B should always be equal. Adding a number to itself K times is the same as multiplying that number by K. On the Horizon A & B are equal for every value tested. Line 80 always prints "Equal." When the benchmark was tested on the IBM-PC with K=1000 and J = .1, the results were: A=99.99904 B=100, "Not equal." For smaller values of K A and B have the same printed values but "Not equal" is printed by line 80. As K gets bigger the differences also grows between A and B.

The AppleII+ using Applesoft (Apple's version of BASIC for real numbers) the benchmark produces similar answers to the IBM-PC. The results of A and B are close, but "Equal" never prints.

Pascal benchmark for floating point

```
Program loop;
  Var
    I,K : integer;
    a,b,c : real;

  Begin
    a := 0.0; b := 0.0;
    Writeln('Upper limit of loop');
    Readln(K);
    Writeln(' Give an addend ');
    Readln(b);

    For I := 1 to K do
      a := a + b;
      c := b * K;
      Writeln(' A= ',a,' C= ',c);
      If a=c then writeln('equal')
        else
          Writeln('Not equal');
```

.These next two lines are later improvements. The curly brackets mean this is a comment.

```
      I := round(a); k:= round(c);
      Writeln('Rounded values: A= ',I,' B= ',k);

  If I = K then Writeln('Equal')
    else
      Writeln('not equal')
  End.
```

Using Apple Pascal the results are:

A	B
1.00001E1	1.0000E1
Not equal	

Changing the number of iterations or the value of the addend gives A and B with equal values sometimes and not equal other times. The two marked lines will round the values. With this improvement A and B look like they are equal.

In Turbo Pascal the rounded values print out the same because they are changed to integers. When the real values are printed A and B are not equal. In Turbo Pascal integers are limited to the values of -32,768 to +32,767. A run-time error occurs if the value of A or B before rounding is outside of the legal integer range.

A programmer must be cautious when using floating point numbers. To compare results use inequalities, round the results to integers or do both. Apples consistently fail the benchmarks when considering speed and accuracy. Inaccuracies in real number arithmetic are common. North Star's implementations are consistently faster and more accurate than Apple7s. Apples may be great for games, but I wouldn't want to bet my accounting pennies on one.

###



=====
Product notes

Z-80 microprocessor chip is made by Zilog Corporation. It is the 8 bit processor that runs the North Star.

CP/M was invented by Gary Kildall and is sold by Digital Research.

Nevada COBOL is marketed by Ellis Computing 3917 Noriega St., San Francisco, CA 94122. It operates under CM/P on both the Apple and the Horizon.

Turbo Pascal is marketed by Borland International 4807 Scotts Valley Drive, Scotts Valley, CA 95066.

Apple Pascal is a trademark of Apple Computers. It incorporates the UCSD Pascal system with Apple extensions for graphics, sound, paddles, and some functions.

Applesoft is Apple Computer's version of BASIC that uses integers and real numbers. There are considerable differences between Applesoft and Apple Integer BASIC.

=====
Daniel D. Stuhlman is the president of BYLS Press, Chicago, IL, has taught computer programming languages at Kendall College, Evanston, IL and has been a North Star Horizon owner since 1979.

BYLS Press publishes two computer personalized books for children.

=====
#

BUGS BUGS BUGS

Hugh Power of Power's Computers, Richmond, CA, one of the most knowledgeable North Star dealers we know, reports a bug in the more recent versions of North Star DOS/BASIC. He suspects that the bug is in the DOS, but it shows up in the crash of BASIC programs. The symptom of the bug is a program which crashes, with an unanticipated, unexplained TYPE ERROR, and a corrupted file.

Hugh has analyzed files on disks sector by sector, and has discovered that DOS writes some particular file sector one sector too high on the disk. Thus a file sector which should be on disk sector eight ends up instead on disk sector nine, or seven ends up on eight, and so forth. The error in writing is always sector to sector and never track to track.

Apparently DOS makes a timing error which results in an incorrect pointer, off by one, or in an incorrect disk write. The problem occurs infrequently, but often enough to be documented, and frustrating to user and dealer alike.

Hugh Power's recommendation, if you ever encounter this problem, is to revert to 5.0 DOS/BASIC, or to transfer your files to CP/M.

#

Look Before You Leap

By Alan H. Nelson

Through a rather complex set of circumstances, the academic department in which I teach has acquired a number of IBM-PC's and an IBM PC-XT. As the departmental hacker, I have been made the person responsible to get them running.

Despite what you may have heard, the news is not all good. PC-DOS (really MS-DOS) is a close imitation of CP/M, with a number of significant improvements, but also a number of devastating drawbacks. I can understand how IBM would be in a position to take advantage of CP/M's successes. It is beyond me how IBM could have allowed its DOS to take giant steps backward.

My message is: **LOOK BEFORE YOU LEAP!**

What's Good about PC-DOS?

The single most useful improvement in PC-DOS is a resident copy command which is far superior to nasty, nasty **pip**, which I despise. Not only is **copy** always available from the DOS prompt, but the syntax is much closer to normal English, i.e. **copy ((from)) source ((to)) target**. Thus to copy a file named temp from drive a to drive b, you would use the following command:

```
A>copy temp b:
```

This is the way CP/M should have done it too.

PC-DOS also has a batch file command infinitely easier to use than CP/M's **SUBMIT.COM**. It supports many UNIX-like commands, including (somewhat primitive) pipes and redirects, and sub-directories.

What's Bad about PC-DOS?

The worst feature of PC-DOS is that it does not permit the recovery of an accidentally erased file. This is a good example of what a friend of mine who uses some of IBM's mainframe programs calls IBM's "fascist" program philosophy. IBM

knows what's best for you, and it's just your tough luck if you make a mistake--it probably means you shouldn't be messing around with computers anyway!

A comparison of file maintenance features is called for here. When you erase a file in CP/M, the program or text stored on the disk isn't actually erased; rather, a single bit is changed in the index area of the disk to indicate that the space occupied by the "erased" file is now available to be written on by new files. The allocation table which records where the file was placed on the disk remains intact. A program can be reclaimed intact on the condition that it has not been written over by subsequent disk writing procedures. Of course, if you erase a file and realize your mistake immediately (this is what usually happens), no disk writes have occurred, and you can get your file back. CP/M itself does not contain a file-reclaim program, but many utility programs have implemented this feature.

I am told by a programmer friend that PC-DOS actually erases the entire allocation table, so that though the file remains on disk, there is no way to recover the information concerning where it resides. Even if you can read the disk directly and thus find the beginning of a file, there is no way (except by examining the entire disk sector by sector) to find a file which has been written to non-contiguous sectors; and even if you do recognize the material as belonging to the same file, it's not always easy to be certain that what you have found is not an old, outdated version of your file.

A further indication of the seriousness of this problem is that my favorite utility program, **POWER!**, has not implemented **RECLAIM or any disk read and write features whatsoever** in its PC-DOS version. Yet, apart from the **COPY** routine, these are the features of **POWER!** I use most in CP/M!

Several additional problems arise with the use of the XT, the 10 megabyte hard disk version of the PC. Any user of a hard disk knows the elation which comes with having millions of bytes free after living

dangerously with floppy disks usually full to overflowing. The elation is quickly cut short, however, at the view of hundreds of filenames flying across the screen after a **dir** command. How do you keep track of all those files, or group them into logical categories?

CP/M has user areas which partly answer the need for segregating files into logical or personal subgroups; PC-DOS has subdirectories. The advantage of subdirectories is that there is virtually no limit to the number you can have on a disk, particularly with subdirectories inside subdirectories. It is also possible to set a **path** so that a program in the main (or **root**) directory can be invoked from a subdirectory.

Simple, eh? Not simple, but stupid, stupid, stupid, because though PC-DOS can find COM or EXE programs in the root directory, those programs cannot find their overlay files!! Therefore it is necessary to copy the overlay files into each subdirectory where that program is to be used, and you may quickly find your precious 10 million byte drive cluttered with multiple copies of overlay files. This problem occurs with WordStar, dBASEII, and all other programs which use overlay files.

IBM of course has its implied answer: don't use any programs which are not specifically written for PC-DOS in the first place. It's WordStar's fault, not IBM's.

Here's another problem: The XT seems perfect for use by a large number of individuals who have a need for access to a large number of programs. The ideal in this situation is to have all the programs on drive A, and to allow each user to put his or her own private disk into the floppy disk drive, which might logically be called drive B.

Most CP/M systems allow the hard disk to be defined as Drive A, or as the major resident drive. Not IBM! The hard disk is drive C, and the floppy drive is drive A. Now PC-DOS has a program called ASSIGN which allows the drive designations to be changed, and it is possible to make DOS think that the floppy disk is Drive B and the hard disk is Drive A. WordStar can be installed, moreover, to understand that both it and its overlay files live on the hard disk. But what happens when you try to RUN a program from WordStar (as I do dozens of times per day)? You will

discover that the drives have not been permanently redefined at all, and programs (including spelling checkers and footnoting programs) crash or lock up, and the system has to be reset.

Once again, of course, it is not IBM's fault, but WordStar's fault, or the user's fault for trying to do things his or her own way rather than IBM's way. In fact, in the (incomprehensible) manual description covering ASSIGN, IBM suggests that you **not use it**, and instead that all drive assignments be specifically invoked.

In contrast to this, I have long felt that one of WordStar's niftiest features is that its RUN command does not require drive designations at all; rather, WordStar looks for a program on the current drive, and if it does not find it there, it looks for it on the drive where WordStar itself resides. Now that is friendly programming! It irritates me no end for IBM to be instructing WordStar in the world of user-friendliness.

Finally, a small and needless insult. CP/M's command for erasing files is ERA. PC-DOS uses two commands, ERASE and DEL. As a person who switches back and forth from CP/M to PC-DOS on a daily basis, it is a great annoyance for ERA to produce an error message in PC-DOS.

Future releases of PC-DOS may overcome some of the shortcomings of the current versions, but until that day comes, the word remains: CAVEAT EMPTOR! If you leave North Star, you may soon come to feel that you have lost the friendliest friend you ever had!

#

FOR SALE

North Star Advantage with 5MB HD and all three operating systems for sale for \$2500 or best offer. This has been used as my demonstrator. It is sold "AS NEW" with all original packing, manuals, and warranties. Might take a Horizon in trade.

Orlo Hudson
P.O.B. 68

New Strawn, KS 66839

Advantage Answers

INSUA:

I have recently purchased a used North Star Advantage computer with 64K memory. It has the 5 M-Bite hard disc and one floppy disc with 340 memory. I would like information on the following items for a small business.

1. Can memory be advanced to 128 K?
2. Can computer be converted to MS-DOS software?
3. Best accounting package for price?
4. Best inventory control package for price?

Any other information you can give us that you feel would be helpful would be very much appreciated. Awaiting your reply.

Respectfully,
Lowell Onstad
Trimetric Specialties, Inc.
37400 "A" Cedar Blvd.
Newark, CA 94560

(The day after this letter came to my attention, I stumbled across the following article in the July, 1984 Polaris, Newsletter of the North Star Computer Society, P.O. Box 311, Seattle, WA 98111-0311. Compass acknowledges its debt to Polaris for such informative articles as this one. --Ed.)

Technical Questions

We got some technical questions after a long quiet spell. Rick Ehlert from Bozeman, Montana provided several questions which will be of general interest.

Q. Using North Star BASIC, how do you make use of the function keys and cursor control keys?

A. Finding out if a key has been pressed is easy using the INCHAR\$(n) function described on page 8-4 of the GBASIC User Manual. The hard part is deciding what to do when a particular key is pressed. The cursor and function keys have no meaning unless the program which is executing

assigns one to them. The cursor and function keys all generate values above 127, which is out of the range of normal ASCII characters. If you have an ADVANTAGE reference manual appendix A-2 tells you the values each key generates with each shift key (SHIFT, CONTROL, CMND). You can find out yourself through the following short program.

```
10 T$ = INCHAR$(0)
20 PRINT STR$(ASC(T$))
30 GOTO 10
```

The cursor keys generate values between 130 and 138 decimal and the function keys (unshifted) generate values between 219 and 233 decimal. Detecting which key has been pressed is a tenth of the battle. Making them do something useful, as they do in InfoManager, is quite a different problem.

You might define strings in the beginning of your program like this:

```
U$ = CHR$(130)
D$ = CHR$(138)
L$ = CHR$(136)
R$ = CHR$(134)
```

And then use them later in a control loop like this:

```
C$ = INCHAR$(0)
IF C$ = U$ THEN GOSUB 1000
  REM PERFORM ROUTINE FOR UP
IF C$ = D$ THEN GOSUB 2000
  REM PERFORM ROUTINE FOR DOWN
IF C$ = L$ THEN GOSUB 3000
  REM PERFORM ROUTINE FOR LEFT
IF C$ = R$ THEN GOSUB 4000
  REM PERFORM ROUTINE FOR
RIGHT
```

Q. Is it possible to move the display on the CRT around using the cursor keys?

A. Using the ADVANTAGE screen as a window onto a larger area is a very attractive idea. I wish North Star had done that with ImageMaker and BusiGraph II. Some word processors provide something like this facility with horizontal scrolling. WordStar, the Electric

BlackBoard, and TurboPascal's editor are some editors I've used recently which do horizontal scrolling. But I haven't seen anything which does something similar for graphic images. It would take quite a bit more knowledge of the ADVANTAGE screen mapping than I have and a lot of fancy programming.

Q. I am considering upgrading to an 8/16 after the recent price reductions. What is a useful configuration? Do I need both GCP/M and GMS-DOS?

A. As I happen to be the only person I know who has an 8/16 board, I can only tell you what my experience has been. First, a minimum hardware configuration is the 8/16 board (it has 64KB on-board) and one additional 64KB although you might find an application which will run in 64KB. Sometimes more memory doesn't get you anything. TurboPascal limits your program source to 64KB no matter how much memory you add; Multiplan can't make use of more than an extra 64KB either. MS-DOS is a larger operating system and programs written for it tend to be larger and fancier also. Next you must buy both GCP/M and MS-DOS. While MS-DOS is running the 8088 cpu, CP/M is running the Z80 as a slave processor and providing I/O and graphics services for the 8088. Be aware that there is no version of North Star Graphics BASIC that runs under MS-DOS or CP/M. There is a non-graphics version of BASIC compatible with North Star's called APCBASIC from American Planning Corporation (4600 Duke Street, Suite 423, Alexandria, VA, 22304) which runs under both CP/M and MS-DOS on the ADVANTAGE. I've never tried it.

Q. Are there any good books about programming with the ADVANTAGE?

A. There are no books at all about programming the ADVANTAGE. If you have the User Manual and the Technical Manual you have all there is specifically about the ADVANTAGE. The same is true of the Horizon, unfortunately.

I would like to say a little more about the 8/16 upgrade for the ADVANTAGE. I got mine about two years ago and found that there was almost no software that worked with it. Initially, all I could find

was the Multiplan spreadsheet program from Microsoft. Since then, a few more programs have appeared and I printed a list of companies that produce North Star-compatible MS-DOS software a few months ago. Don't rush out and buy the 8/16 upgrade unless you have a real need for faster execution or more memory space and make sure that the program you need is available for the ADVANTAGE.

#

Commentary

Terry Hough
P. O. Box 13309
Greensboro, NC 27405

On accessing the HDOS File Manager from North Star BASIC. The following two routines allow you to directly communicate with the HDOS File Manager and to use its built in routines to augment your basic programming.

You've seen this method used to load a machine language routine before and I think you will find the programs themselves self-explanatory. I do suggest you review section 5 of your HDOS manual for the operations the file manager can perform for you.

FMLX will allow you to directly begin execution of a Type 1 file from your basic program. Here it is set to start TSS/C or you can prompt for the desired file.

FMSDV allows you to change HDOS's default account. There are many uses for that, such as changing to another turnkey software packagows you to change HDOS's default account. There are many uses for that, such as changing to another turnkey software package stored in a different account on the hard disk from a overall system menu.

I've found these routines very useful and hope you will too. If you have a suggestion or question, drop me a note.

#

Disk Controller PROMs

By Joe Maguire

Of the three PROM chips on the North Star double density disk controller board, two remain a mystery.

The bootstrap program PROM (DPGM) can be read by examining the memory area from E800 to E8FF (standard version) with the Monitor program. A disassembly of this PROM was given in Compass, Vol. 2, No. 1.

The remaining two, DSEL and DWE, do not have their outputs tied to the data bus and cannot be read with the Monitor. They are programmed, however, and do have a purpose. Their contents, and function, will be the subject of this article.

What's a PROM?

PROM stands for Programmable Read Only Memory. The ones used on the controller board are of the TTL variety sometimes referred to as "bipolar." These differ from the NMOS type, such as the 2708 and 2716, by internal construction.

North Star uses two types on the board. The DPGM PROM is a 256x8 or 2048 bit type 74S471. This means that its internal arrangement consists of 256 eight bit bytes. That's easy to see when you dump it. It looks just like any other memory dump.

The DSEL and DWE PROMS are 256x4 or 1024 bit type 74S287. (North Star lists these as 82S129s but that is a Signetics house number.) The internal arrangement is 256 four bit nibbles. Think of a nibble as half a byte. (Honest!)

The true addresses of all these PROMs start at zero and go to FF hex. The circuitry on the controller board shifts them to apparent other addresses.

A PROM is really a do-it-yourself logic block. Some standard TTL ICs, such as the address decoder type 74LS138, are really PROMs in disguise. The 74LS138 has three input lines and eight output lines. Three to eight works nicely when you want to convert binary to decimal. But what type do you select when you have eight input lines and want four outputs? No standard type will give you that.

There are just too many possible combinations for an IC manufacturer to attempt to decide what should be "standard." Instead, they offer you a blank IC and tell you to program it yourself.

A blank PROM is filled with all zeros. The user programs the zeros to ones, on a selected basis, to get the functions he wants. The programming is done by special devices which can control the currents (up to half an amp) and voltages (up to 30V) required.

The DSEL PROM

DSEL stands for "disk select." It is really the PROM which determines the memory address at which the controller board will respond. It decodes four addresses out of the possible 256 and, based on the address byte, directs the controller to take various actions.

One way of looking at the required list of actions is by means of a "truth table." When an item in the table comes up true, then some action should be taken. If the item is not true, don't do anything. True can be defined as either a logic high or low. For our table we are going to define true as low or a zero.

Table 1. is the truth table for DSEL. Some explanation of terms is in order:

PROM Select - When this is true, the bootstrap PROM output is enabled. The CPU reads the PROM on startup. After that it is not used.

MUX Enable - MUX stands for multiplex. The multiplexers are used to put the status byte and read data on the data bus.

Orders - Various circuits within the controller must be given instructions from time to time (read, write, step, etc.) When this goes true, the controller knows a command is coming up.

Stall - The disk is a much slower device than the CPU. From time to time the CPU must be held in a wait state until the disk

completes an operation. True on this line makes the CPU wait.

The DWE PROM

DWE stands for "disk write enable." The DWE PROM controls the various functions required for proper writing to the disk. Writing is the most critical of disk operations. Various parameters must be carefully controlled if data integrity is to be maintained. For example, when writing on the innermost tracks, write precompensation must be used because of the slower linear velocity of the disk. DWE takes care of this.

The DWE PROM is not affected by locating the controller board at a different RAM address. Therefore, if a special PROM set is ordered to relocate the disk, DWE need not be changed.

The inputs to DWE are a combination of signals from onboard the controller and not an address as in the case of DSEL. However, the PROM responds as if it were

RAM address high byte	Stall (CPU wait state)--- Controller Orders--- MUX enable----- PROM select-	Hex value
E8	0 1 1 1	7
" "	1 1 1 0	E
" "	1 1 0 1	D
" "	1 0 1 0	A

Table 1.

PROM "DSEL" truth table. (standard address)
0 = True. All nibbles at addresses other than those shown above contain value F hex.

From the table it can be seen that when the CPU is addressing the E8 block, only the PROM is enabled. When the EA block is addressed, only the orders register is enabled etc.

an address and selects the various functions accordingly. The input signals are not as meaningful to the user as those of DSEL and will not be elaborated upon here. Refer to the disk controller schematic for more information.

Table 2. is a dump of the DWE PROM. An input example is given to show what effect it has on the output. Again, refer to the schematic for information about the various output lines.

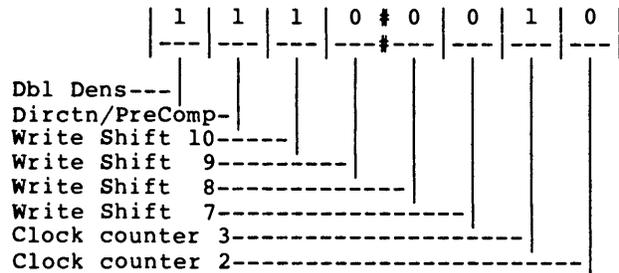
#

00	7 1 7 F	7 1 7 F	7 1 7 9	7 1 7 9
10	7 1 7 F	7 1 7 F	7 1 7 9	7 1 7 9
20	7 1 7 F	7 1 7 F	7 1 7 9	7 1 7 9
30	7 1 7 F	7 1 7 F	7 1 7 9	7 1 7 9
40	7 1 7 F	7 1 7 F	7 0 7 9	7 0 7 9
50	7 3 7 F	7 3 7 F	7 1 7 9	7 1 7 9
60	7 1 7 F	7 1 7 F	7 0 7 9	7 0 7 9
70	7 3 7 F	7 3 7 F	7 1 7 9	7 1 7 9
80	1 F 1 F	1 F 1 F	7 9 7 9	7 9 7 9
90	7 F 7 F	7 F 7 F	7 9 7 9	7 9 7 9
A0	1 F 1 F	1 F 1 F	7 9 7 9	7 9 7 9
B0	7 F 7 F	7 F 7 F	7 9 7 9	7 9 7 9
C0	1 F 1 F	3 F 3 F	7 9 7 9	7 8 7 8
D0	7 F 7 F	7 F 7 F	7 B 7 B	7 9 7 9
E0	0 F 0 F	1 F 1 F	7 9 7 9	7 8 7 8
F0	7 F 7 F	7 F 7 F	7 B 7 B	7 9 7 9

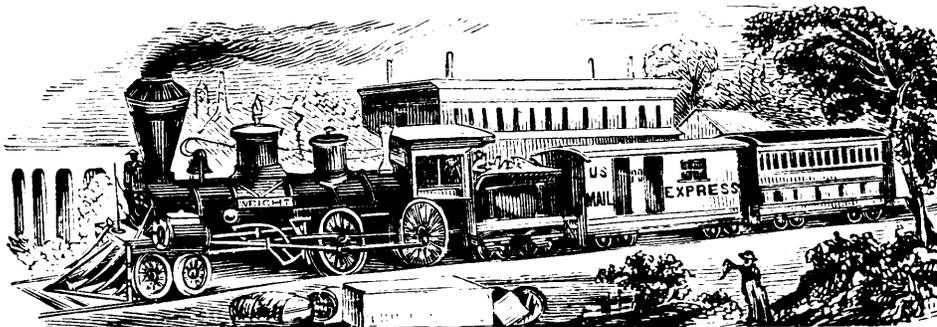
Table 2.

PROM "DWE" dump. Left column is address. Display is sixteen nibbles per line in hex.

"Address" is composed of the following:



The example shows "address" E2. From table 2., the output is 0, all outputs enabled.



A File-Size-Changing Program

By Saul G. Levy
2555 E. Irvington Rd., Lot 47
Tucson, AZ 85714
(602) 889-7753

FILESIZE may not be needed very often, but it is very handy when you need to change a lot of file sizes. You will need 34K of memory and only one diskette drive to RUN it, but you do need two drives to use the CF (Copy File) utility to copy the old data into the new files. You must use Release 5 or later software (the file CREATE statement does not occur in Release 4 or earlier BASICS).

Has This Ever Happened to You?

It is now the beginning of November. You get a call from the church telling you that all of the accounting files are full (the automatic file creation did not make them large enough). No problem, you tell the minister. Well, changing the sizes of a lot of files is a problem!

TWO WAYS to Handle File Sizes

The main claim-to-fame of CP/M is its automatic file allocation. If you need (and have) an extra sector, then CP/M will allocate it to whichever file needs it at the time. North Star files are of fixed length and cannot be changed easily. You can increase the size of the last file (if you have the room), but how many times does that happen, anyway?

Program FILESIZE to the Rescue!

This program will accept a constant, size-factor number which will change the file sizes in one of TWO WAYS (natch!) depending on the file Type being changed (one Type at a time to give you some choice):

1. All Type 0-2 files can be increased with a positive size factor or decreased with a negative size factor. The size-factor number must be an integer and represents the size change in BLOCKS (256 bytes per block).

2. All Type 3-127 files can be changed by MULTIPLYing by the size factor. This number must be positive and can be a fraction.

How About Some Examples!

Usually you will want to increase the sizes of Type 3 data files to accommodate more data. If you enter a size factor of 2, all Type 3 files will be DOUBLED in size. If there is no room for this, try 1.7 instead. If you want to DECREASE the sizes, entering a .5 will HALVE the file sizes. Likewise, a 3 will TRIPLE the sizes and a .333 will cut each file to one third of its original size.

If you want to increase the size of a BASIC program file (Type 2), you can enter a 1, 2, 3, or 4 for minor changes and see the new files increase by 1, 2, 3, or 4 blocks. Use a minus sign for decreasing the sizes: -1, -2, -3, or -4. You could use this mode to change the size of every BASIC program file if you want to add some code to every program in a package. This is needed less often than the previous case.

Double-density file sizes are EVEN numbers of blocks. If an odd size occurs, it will be CREATED at the next higher even size, i.e., if the new file is five blocks long, a six-block file will be CREATED.

Using Program FILESIZE

FILESIZE uses code from my directory utility program DIRUTIL (see Vol.II, No.4, p.32). Your BASIC must be able to CREATE files (Release 5 or later). Like DIRUTIL, you must name the directory file on the old diskette. To do this on drive 1, enter DOS and type:

```
CR DIR 4 0 or CR DIR 8 0
```

(use 'DIR,2' for Drive 2)

The first one is for single-density, the second for double/quad. The name DIR (or whatever) may be used on all of your diskettes, or you can number them or use a different name on each to give a better idea of the contents (for example: FILES, FILES2, ACCOUNT, PAYROLL, etc.). Note that the directory file was not TYPed. You must use the '0' to create a directory file which starts at disk address zero.

Put the OLD diskette in any drive and RUN Program FILESIZE (it is safest if you write-protect this diskette until you are done).

The first thing you will enter is the directory name (DIR or whatever) along with a drive number (if needed, Drive 1 is the default). The directory will be read and the instructions will be listed. You may then remove the old diskette.

Load your NEW diskette into any drive. Normally this will be an INitialized diskette which is empty. After the instructions you will be asked for the file Type you wish to change (one at a time to give you some choice). Then you will enter the constant size factor and finally, the drive number where you loaded the new diskette. Just hit RETURN for the default of Drive 1.

Every OLD file of the Type you specified will be CREATED on the NEW diskette in the drive you are using. The program will show you the old and new sizes in blocks. If any of these files already exist on that drive, a message will be printed instead of the sizes and that file cannot be CREATED or its size changed. This makes the program harmless if you forget to replace the old diskette!

NOTE that this will only CREATE the new files, it will not copy the file data to the new files! Use the CF (Copy File) utility for that job as long as the new files are LARGER than the old ones (CF will not copy an old file into a new file too small to hold it, sorry about that).

You may manually CREATE any file by hand if you do not like the sizes FILESIZE gives you. You should CREATE them on an empty diskette BEFORE you RUN this program so you will have enough room for the other files that FILESIZE will CREATE. Otherwise, you will have to DESTROY these files before recreating them by hand, and may have to COMPact the diskette before you can CREATE all of them.

Two Minor Problems

If you run out of room on the new diskette before all of the files you need are CREATED, you will have to change the size-factor number until they all fit. Be sure to DESTROY the incorrect new files BEFORE starting the program again, otherwise, nothing will happen! You can reINitalize that diskette, but do not INitalize the old diskette by mistake if it is still loaded! If you CREATED any files of a special size, do not DESTROY them! You could also use multiple diskettes to hold the new files.

If you should CREATE a new file of the old diskette's directory name (usually of Type 0), the CREATED file will not be the new diskette's directory name! BASIC will CREATE this file starting in Sector 4 or higher on the diskette, NOT in the directory area.

Only One Major Problem!

Just because you increased the file sizes and copied the file data over to the new files does not mean that the software which uses those files can use the extra sectors! My programs keep track of the file sizes and tell you when they are almost full. Other programmers fix the number of records which their files can handle. I hope they warn you when the files are almost full, but you will have to find out where they store the maximum number of records and change each of them to the new number. There is no easy way to tell you how to do this. I do recommend that you always use the NOENDMARK statement to prevent clobbering anything else which may be stored just after the number of records!

You may not need to change the number of records; just the file sizes. If you do not know how to find and change the maximum record numbers, do not even try it! There is no easier way than this to destroy a data file and drive yourself crazy! Be sure to keep the old diskette until you are sure everything is O.K.

Modifications

A completeness variable is used to shorten the reading of the named directory. It will stop reading after FIVE EMPTY FILE NAMES in a row have

occurred (Line 420). Change it to suit (up to 127) as it is not harmful in any way, but file names which have not been read will not be CREATED. Just change the value of C and reRUN. Change the clear screen string in Line 250 to whatever you need (an ADM-3A here).

A variable and line number cross-reference listing is included if you wish to change anything else. The program to do this is by Larry Hudson (see Dr. Dobb's Journal, No. 49, October, 1980, p. 32). It is the best cross-reference program I have seen. I can supply single- or double-density versions if anyone wants a copy.

Copyright Notice and Distribution Rights

You may remove all REMarks and unnecessary spaces, which take up over 60% of this program, but do not remove my copyright notice in Line 40. This program may not be sold in any way or in any form by anyone (except INSUA) including user group diskettes which have any fee connected with them. I do authorize free distribution of this program ONLY IN ITS COMPLETE FORM to anyone you wish.

A Plug for a Faster Way to Copy Files

One thing CP/M does very slowly (at least on North Star diskette drives) is file

copying. I can guarantee that CF is much faster than PIP at copying files. The main problem is correctly typing those strange file names!

I have just finished a Smart Copy File program (SCF) which will copy files very rapidly with the CF utility (Copy Disk, CD, is also usable). SCF does need to be configured for your system although I can do this for you if you tell me what your system consists of (terminal name, clear screen command, direct cursor addressing command, memory addresses available, and which Release software you use most often; the later the better). (Release 4 DOS has a serious limitation: The built-in CF cannot CREATE a new file! Regular backups to old files work fine.) You must test one thing on your system even if I have configured your version. I offer full support of my software and include 14 pages of documentation that you can print yourself (if you do not have a printer, I will print it for you).

SCF is available for \$20.00 if you send me one of your diskettes (\$25.00 if I use one of mine). The cross-reference program is in the public domain and is FREE (tell me how many columns and how many lines your terminal has, and the same memory and Release information as above). Please include an extra \$3.00 for postage and handling with each order (my address is at the beginning of this article).

```

10 REM PROGRAM FILESIZE, A FILE SIZE CHANGING PROGRAM
20 REM WRITTEN BY SAUL G. LEVY, TUCSON, ARIZONA, NOVEMBER 14, 1982
30 REM LAST CHANGED MARCH 11, 1984
40 REM (C) COPYRIGHT SAUL G. LEVY, 1982, 1984
50 REM
60 REM
70 REM CREATE A DIRECTORY NAME ON THE OLD DISKETTE WITH THE FOLLOWING COMMANDS-
80 REM
90 REM BYE (LEAVE BASIC AND ENTER THE DOS)
100 REM CR DIR 8 0 (FOR DOUBLE/QUAD-DENSITY ON DRIVE 1,
110 REM OR USE DIR,2 FOR DRIVE 2, ETC.)
120 REM CR DIR 4 0 (FOR SINGLE-DENSITY ON DRIVE 1)
130 REM JP E04 (RETURN TO BASIC WITH THIS PROGRAM INTACT, A RELEASE
140 REM 5.200 ADDRESS)
150 REM NOTE THAT 'DIR' CAN BE ANY LEGAL FILE NAME OF ANY TYPE (TYPE 0 HERE)
160 REM
170 REM
180 REM ERROR SET
190 ERRSET 1000,20,29
200 REM
210 REM DIMENSIONS
220 DIM D(127,15),N$(10),Z9$(17)
230 REM
240 REM CLEAR SCREEN (ADM-3A)
250 A$=CHR$(26)
260 ! A$,TAB(30),"PROGRAM FILESIZE"
270 !
280 !
290 REM
300 REM ENTER DIRECTORY NAME
310 INPUT "ENTER DIRECTORY NAME (<,DRIVE # IF NOT DRIVE 1): ",D$

```

```

328 REM IS DIRECTORY NAME VALID? T1 IS THE FILE TYPE
340 T1=FILE(D#)
350 IF T1<>1 THEN 420
360 !
370 ! "DIRECTORY NAME IS INCORRECT, CTRL-G WILL REENTER SAME ONE"
380 !
390 GOTO 270
400 REM
410 REM SET FLAG
420 C=5 \ REM COMPLETENESS FLAG, 127=READ THE WHOLE DIRECTORY,
430 \ REM
430 REM
440 REM N IS THE NUMBER OF VALID ENTRIES-1
450 REM N IS THE NUMBER OF VALID ENTRIES-1
460 N=N-1
470 REM
480 REM OPEN DIRECTORY FILE, Z IS THE SIZE OF THE FILE IN BLOCKS
490 OPEN #0/T1,D#,Z
500 !
510 ! "READING THIS DIRECTORY'S FILE NAMES, SIZES, AND TYPES"
520 REM
530 REM
540 REM READ DIRECTORY, D(N,15) WILL ALWAYS BE A BLANK SO DO NOT READ IT, IF
550 REM D=32 THEN THIS ENTRY IS EMPTY, C1 IS THE NUMBER OF EMPTY ENTRIES
560 REM SKIPPED OVER SO RESET IT TO 0 EACH TIME A VALID ENTRY IS READ
570 FOR I= 0 TO 16&Z-1
580 READ #0/I,16,&D
590 IF D=32 THEN 660
600 C1=0
610 N=N+1
620 D(N,0)=D
630 READ #0,&D(N,1),&D(N,2),&D(N,3),&D(N,4),&D(N,5),&D(N,6),&D(N,7),&D(N,8)
640 READ #0,&D(N,9),&D(N,10),&D(N,11),&D(N,12),&D(N,13),&D(N,14)
650 GOTO 680
660 C1=C1+1
670 IF C1=D THEN EXIT 710
680 NEXT I
690 REM
700 REM CLOSE DIRECTORY FILE
710 CLOSE #0
720 REM
730 REM PRINT INSTRUCTIONS
740 REM
750 ! A#,TAB(23),"FILE SIZE CHANGING INSTRUCTIONS"
760 !
770 ! "LOAD AN INITIALIZED DISKETTE INTO ANY DRIVE (YOU CAN REMOVE THE
780 ! "DISKETTE JUST READ). YOU MAY USE A DISKETTE WITH OTHER FILES ON IT."
790 !
800 ! "THE OLD FILE NAMES CAN BE CREATED WITH A DIFFERENT SIZE ON THE NEW"
810 ! "DISKETTE LOADED ABOVE. ONLY ONE FILE TYPE AT A TIME CAN BE CREATED!"
820 ! "THE OLD SIZES WILL BE CHANGED BY A CONSTANT SIZE FACTOR."
830 !
840 ! "YOU SHOULD CREATE ANY FILES OF SPECIAL SIZE BEFORE RUNNING THIS"
850 ! "PROGRAM! BE SURE TO TYPE THEM! SUCH FILES WILL NOT BE CHANGED."
860 !
870 INPUT "HIT RETURN TO CONTINUE ",Z$
880 !
890 ! "FILES OF TYPE 0-2 WILL ADD OR SUBTRACT THE CONSTANT SIZE FACTOR TO THE"
900 ! "OLD FILE SIZES (-10, -5, -1, 0 FOR SAME SIZE, 1, 5, 10, ETC.). THIS"
910 ! "NUMBER IS IN BLOCKS (OF 256 BYTES EACH)."
920 !
930 ! "FILES OF TYPE 3-127 WILL MULTIPLY THE CONSTANT SIZE FACTOR TIMES THE"
940 ! "OLD FILE SIZES (.333, .5, 1 FOR SAME SIZE, 2, 3, ETC.). POSITIVE"
950 ! "NUMBERS ONLY! A CHECK IS MADE IF AN EXCESSIVE SIZE FACTOR (<1 OR >2)

```

```

960 ! "IS ENTERED FOR TYPE 3-127 FILES ONLY."
970 !
980 ! "IF ALL OF THE NEW FILES WILL NOT FIT, YOU WILL HAVE TO USE A DIFFERENT"
990 ! "SIZE FACTOR OR MULTIPLE DISKETTES (THE INCORRECT FILES WILL HAVE TO BE"
1000 ! "DELETED BY HAND). IF YOU REINITIALIZE, BE SURE TO CHECK FOR THE"
1010 ! "CORRECT DISKETTE BEFORE IT IS TOO LATE!"
1020 !
1030 ! "USE PROGRAM CF (COPY FILE) TO ACTUALLY COPY THE CONTENTS OF THE OLD"
1040 ! "FILES OVER TO THE NEW FILES. NOTE THAT ANY GO ADDRESSES WILL BE"
1050 ! "INCORRECT. THESE WILL BE FIXED BY THE CF PROGRAM."
1060 !
1070 INPUT "WHICH TYPE OF FILE DO YOU WANT (0-127): ",T
1080 IF T<0 OR T>127 OR T<>INT(T) THEN 1080
1090 !
1100 INPUT "WHAT SIZE FACTOR DO YOU WANT TO USE ON ALL FILES: ",F
1110 IF T<3 AND F=INT(F) THEN 1190
1120 IF T<3 THEN 1110
1130 IF T<3 THEN 1110
1140 IF F<=0 THEN 1110
1150 IF F>=1 AND F<=2 THEN 1190
1160 ! "IS YOUR FACTOR CORRECT: ",F," : (Y OR N): ",
1170 INPUT " ",C$
1180 IF C$<>"Y" THEN 1110
1190 !
1200 INPUT "DRIVE # FOR OUTPUT DISKETTE (CR = DRIVE 1): ",D$
1210 IF LEN(D$)=0 THEN D$="1"
1220 IF LEN(D$)>1 THEN 1200
1230 !
1240 ! "HERE WE GO!!!"
1250 !
1260 !
1270 !
1280 ! "FILE NAME OLD NEW <==SIZES IN BLOCKS"
1290 !
1300 REM
1310 REM MAIN LOOP
1320 REM
1330 FOR I= 0 TO N
1340 REM
1350 REM CHECK FOR UNWANTED TYPE OF FILE
1360 IF D<1,12><T AND D<1,12><T+128 THEN 1660
1370 REM
1380 REM CREATE STRING OF FILE NAME CHARACTERS FOR EACH FILE WANTED
1390 N$=""
1400 GOSUB 1960
1410 REM
1420 REM NEW FILE ON DRIVE 2
1430 N$=N$+" "+D$
1440 REM
1450 REM DETERMINE OLD SIZE, IF DOUBLE-DENSITY CONVERT INTO BLOCKS
1460 S=D<1,10>+D<1,11>*256
1470 IF D<1,12>>127 THEN S=S*2
1480 REM
1490 REM DETERMINE NEW SIZE, IF ILLEGAL, CHANGE TO 1 BLOCK OR INTEGERIZE IT
1500 IF T<3 THEN S1=S+F ELSE S1=S*F
1510 IF S<1 THEN S1=1
1520 S1=INT(S1)
1530 REM
1540 REM CREATE FILE ON DRIVE 2 ONLY IF IT DOES NOT CURRENTLY EXIST
1550 IF FILE(N$)=-1 THEN 1620
1560 REM
1570 REM FILE ALREADY EXISTS ON DRIVE 2
1580 ! "FILE ",N$,TAB(16),"ALREADY EXISTS ON DRIVE ",D$
1590 GOTO 1660

```

```

1600 REM
1610 REM SHOW FILE DATA
1620 ! N*,TAB(10),%SI,S,S1
1630 REM
1640 REM CREATE NEW FILE ON DRIVE 2
1650 CREATE N*,S1,T
1660 NEXT I
1670 REM
1680 REM NORMAL FINISH
1690 !
1700 !
1710 ! "NORMAL FINISH"
1720 !
1730 !
1740 INPUT "DO YOU WANT TO CREATE MORE FILES FROM THIS DIRECTORY (Y OR N): ",C*
1750 IF C*="Y" THEN 1860
1760 END
1770 REM
1780 REM ERROR MESSAGE ROUTINE
1790 REM
1800 RESTORE 1810
1810 DATA "INVALID ARGUMENT","DIMENSION","OUT OF BOUNDS","TYPE","FORMAT"
1820 DATA "LINE NUMBER","FILE","HARD DISK","DIVIDE BY ZERO","SYNTAX","READ"
1830 DATA "INPUT","ARGUMENT MISMATCH","NUMERIC OVERFLOW","STOP","LENGTH"
1840 FOR I= 1 TO Z*
1850 READ Z9*
1860 NEXT I
1870 !
1880 !
1890 ! Z9*," ERROR IN LINE",Z8
1900 !
1910 END
1920 REM FINISHED
1930 REM
1940 REM CONVERT FILE NAME INTO STRING N* ROUTINE
1950 REM
1960 FOR J= 0 TO 7
1970 IF D(I,J)=32 THEN EXIT 2000
1980 N*=N*+CHR*(D(I,J))
1990 NEXT J
2000 RETURN
2010 REM FINISHED
2020 REM END OF PROGRAM FILESIZE

```

	LINE NUMBER	CROSS REFERENCE
	270 -	390
	420 -	350
	660 -	590
	680 -	650
	710 -	670
	1060 -	1750
	1080 -	1090
	1110 -	1130 1140 1180
	1190 -	1120 1150
	1200 -	1220
	1620 -	1550
	1660 -	1360 1590
	1800 -	190
	1810 -	1800
	1960 -	1400
	2000 -	1970

VARIABLE	CROSS REFERENCE FOR FILESIZE									
A*	250	260	750							
C	420	670								
C*	1170	1180	1740	1750						
C1	680	660	670							
D	580	590	620							
D*	310	340	490	1200	1210	1220	1430	1500		
D()	220	620	630	640	1360	1460	1470	1970	1980	
F	1110	1120	1140	1150	1160	1500				
I	570	580	680	1330	1360	1460	1470	1620	1660	
	1840	1860	1970	1980						
J	1960	1970	1980	1990						
N	460	610	620	630	640	1330				
N*	220	1390	1430	1550	1580	1620	1650	1980		
S	1460	1470	1500	1620						
S1	1500	1510	1520	1620	1650					
T	1080	1090	1120	1130	1360	1500	1650			
T1	340	350	490							
Z	490	570								
Z*	870									
Z8	190	1890								
Z9	190	1840								
Z9*	220	1850	1890							

Zips & Names

In accordance with the announcement in the last issue of Compass, we list here current members of INSUA in zipcode order, publishing only zipcodes and names. The purpose of this listing is to assist INSUA members in getting in touch with one another by consulting telephone books. The privacy of members who have chosen not to have their names listed in local telephone books should remain protected.

00626	RAFAEL A. ALCOCER	07054	WARNER HAUER	11203	MARCELINO F. SIERRA, PH.D.	15146	JOHN P. THORNTON SR., CPA
01002	WALTER KOHLER	07061	LINDA GABRIEL	11214	ROBERT FALK	15641	SAMUEL VINCENT
01201	JOHN G. BURNS	07094	JEROME HAGGART	11236	JERALD ABRAMS	15668	HARVEY HECKER
01302	LEVY WINER, P.C.	07306	FRANCIS X. DORRITY	11354	EDWARD HO	15801	J. THOMSON
01545	DAVID STEFFEN	07430	TIM OBERLE	11419	HARRY RIES	15901	PAUL CARPENTER, CPA
01581	RAYMOND ELIOTT	07446	WILLIAM T. KELLY, CPA	11434	LAWRENCE J. CULLEN	16335	LEONARD HUNTER
01588	BARRY P. GREEN	07470	BRUCE T. BUBELLO	11434	LAWRENCE J. CULLEN	16801	DR. RONALD B. STANDLER
01720	APPLIED MEASUREMENTS, INC.	07481	KAMAL G. BAHRRI	11542	WILLIAM GRAY	17214	R. MACK
01746	ROBERT ROSS	07601	WARREN J. KAPS	11550	DR. E.J. AGNELLO-CHIRMAN	17331	DR. IVAN L. FILLMORE
01757	BRUCE DE RIENZE	07621	FRANK GEIGER	11550	GERALD J. STEIN	17401	LAWRENCE N. DIXON
01778	DONALD HAY	07621	JAMES LAUSER	11552	MURRAY GILBERT	17601	JOE MALAK
01904	ROBERT GILFORD	07666	DENNIS GERMAN	11576	EVAN KATZ	17901	RUSSELL E. MUENCH, JR.
01915	RONALD J. SUBLER	07670	RENO A. DEL BEN	11693	STEPHEN A. FARMER	18103	GORGE CONLY
02148	NICHOLAS J. STRAFELLO	07748	WILLIAM BRYANT	11722	LEONARD SCHMIRL, DR	18105	E.A. LINEBERGER
02158	DAVID YATES/AHTI	07834	ROBERT H. MARTIN	11725	JOHN C. PEREIRA	18328	CHARLES PROHASKA
02173	MERTON KAHNE	07871	HOWARD A. BROOKS	11726	ALAN SHAW	19072	STEPHEN FLINK
02178	EDWARD GANSHIRT	07901	J. WM. WOYTHALER	11743	ROY STOCKDALE	19087	THOMAS CONTI
02192	ROGER W. SINNOTT	07960	G.F.F. HAYS	11743	BRUCE W. HARKNESS	19104	DONALD B. MILES
02543	MORTON MASER	08003	GREGORY D. HICKIN	11776	HARVEY ROSENBLATT	19115	SUNSHINE
02748	SILVINO C. FERREIRA	08028	RICHARD SEIDER	11780	DR. DOROTHY WINTER	19144	JAN COMANDA
02857	GEORGE C. ARNOLD, JR.	08033	RONALD L. MARGESON	11793	DENNIS CASHT'N	19355	ROBERT W. APPEL
02871	ERNEST J. LEVESQUE	08060	JERRY C. SEKERKE	11794	NICHOLAS CARNEVALE	19382	W.R.R. BASHAW
02901	WILLIAM D. COUNTS	08109	JOHN R. GRANT	11797	DR. E. DUEK	19446	WILLIAM T. JOHNSON
02906	WILLIAM L. MAYER	08323	WILLIAM SCHABACKER	11901	DOUGLAS A. RANDALL	19567	CLIFFORD B. SHERMAN
03049	JOHN A. GALLANT	08822	JOHN T. KALINOWSKI	12180	RICHARD HOGAN	19901	RAY SCARBOROUGH
03222	DOROTHY CATE	08857	LEE DANIEL QUINN	12590	JAMES M LOMMEL	20007	CHRISWELL D LANGHORNE DR
03282	KING LUMBER CO.	08903	GENE A. GIACOMELLI	12590	NEAL CHRISTENSEN	20007	GRAHAM ATKINSON
03824	RUSS HALEY CO.	08904	DAVID ROSENBERG, PH. D.	12953	HARVEY FACTOR	20011	FRANCIS C. HABER
03827	LLOYD H. DIXON, JR.	09011	LTC DAVID S. KLUNK	13041	WALLACE B. WATSON	20011	LOUIE BLALOCK
03840	DAVID L. RICHARDS	09055	OLIVER C. STOKES	13057	WIN BABCOCK	20601	AJ ELECTRONICS
03857	DAN HELSEY	09757	REX M. VENATOR	13214	TOBY NADEL	20686	DWIGHT A. JOHNSON
03902	WES BARBOUR PHOTOGRAPHICS	10001	MARK & PAMELA SHEINGORN	13760	PAUL F. CUSHING	20740	ROBERT D. PARR
04558	FREDERICK H. RAAB	10007	ALAN J. STOPER	13850	HARVEY ROEHL	20767	JOHN JACKSON
05401	MICHAEL CRISAPULLI	10011	JOEL KREMSDORF	13903	WILLIAM H. CONNOR	20770	ROBERT J. SPEAR
06089	ROLAND MALBOEUR, E.A.	10011	TERRIE BATDORF	13905	JOHN B. WADE, INC.	20770	DONALD M. COAKLEY, JR., CPA
06095	JOHN H. ADEE, CLU MANAGER	10017	HARRY DEWIG	13905	DR. NICOLAY P. TIMOFGEFF	20782	JAMES E. INGE
06242	WARREN C. CAMPBELL	10025	J.J. DANNENBERG	14036	SCOTT S. CARTER	20816	RICHARD BERG
06268	DR. SCOTT W. BROWN	10028	GABRIEL KATONA	14051	JAMES H. GUSTAFSON	20816	NORA M. TAYLOR
06404	MICHAEL E. BOUTIN	10032	PHILIP PINCHES	14072	ANTHONY J. NOWAKOWSKI, INDIAN	20816	HOLLIS B. VAIL
06442	DOUGLAS S. ATWOOD	10036	H.S. MIDDENDORF	14132	ROBERT APPEGATE	20850	ALAN H. CZARAFATA
06470	WILL TOLL	10038	TED BLANK	14150	RICHARD LOCKWOOD	20850	R. BRODIE III
06470	IRVIG C. ZACHER E.A.	10128	KATHLEEN L. KELLY	14150	ARTHUR KURTZ	20854	BURTON ANDREWS
06470	ROBERT MUNGER	10271	W.E. ERICKSON, JR.	14150	ROBERT G. DISPENZA	20878	JACK HEYTEM
06489	GEORGE DOMBROWSKI	10276	BILL SHOR/VENUS RECORDS	14201	PETER JAFFE	20895	DAN LURIE, PH. D.
06497	MR. R. DONALD REED	10468	PETER MELZER	14216	ALAN GROSS, D.D.S.	20895	WILLIAM E. WAGNER
06512	FRED LINTHICUM	10471	PHILIP W. BRANDT	14216	SPECTRUM SLIDES	20901	RON RASPET
06525	HENRY B. ROTHBERG	10475	WALTER W. STUMBERGER	14216	DAVID ANDERSON	20901	KENNETH W. PARKER
06606	SANTIAGO ESCOBAR, M.D.	10512	JAY JUDELL	14221	GREGORY J. WHITE	20902	R. ROWLAND
06611	ROBERT CARBONE	10514	HAL SKURNICK	14221	ROBERT J. SCHUDER, M.D.	20904	WILLIAM A. TYNAN
06798	JOSEPH A. POSTER	10514	ROBERT PASTOVSKY	14226	ROBERT ACKERHALT	20904	ROBERT J. COATES
06830	A.J. REIG	10541	NORMAN F. BRAY	14226	EVAN J. MORRIS	20910	GORDON RIEL
06851	JAMES M. MCLERAN	10549	L. MARK RUSSAKOFF	14226	MARK N. TAYLOR	21001	HAROLD J. ASHCRAFT
06852	DR. A.J. GUSTAVSEN	10550	K. TERKELSEN	14469	JOHN A. BRYANT	21043	ALLAN HOBBY
06854	ALAN KEYWORTH	10583	ERIC LINDOW	14580	KENNETH L. CLUM	21146	STEPHEN R. TROY
06877	ANDR MCHOSE, CONSULTANT	10803	FREDERICK J. SKINNER	14612	DAVID STEWART	21201	MARK A. KURZMACK
06880	DONALD R. GORDON	10956	JOEL WEISER, M.D.	14612	GORDON LINK, MR. LIGHTING	21207	STANLEY BARON
06883	PETER SIMMONS	10977	HARRY SHAMES	14614	HARVEY M. NUSBAUM	21209	STANFORD LAMBERG
06903	HARRY J. BERKLEY	10977	PHILIP BOND	14618	ERWIN RAHN	21209	ELI FREEDMAN
07006	ROBERT L. BELL	10983	EDWARD H. MCAVOY	14621	GLENN M. HELLMAN	21215	HECKER DISTRIBUTORS
07010	ARTHUR SHEBAR	11050	HARRY ZAM	14623	ROBERT ROSENBLUM	21217	C.J. PENNINGTON
07042	FREDERICK D. SIBLEY	11201	LAWRENCE CHERTOFF	15044	MICK SHEFLER	21229	JAMES SMITH
				15068	DAVID REVILLA	21401	STODDARD "TED" KNOWLES
						21401	JOHN REMIAS
						21502	JAMES J. GRABENSTEIN

21701	FRED ABELES	30323	FELICIA A. TURNER	37215	JAMES ANDERSON	47401	B & L COMMUNICATIONS, INC.
21701	DAVID E. MACDONALD	30338	ROGER A. BEIGH	37220	THOMAS M. BROWN	47421	DAVID B. DROSTE
21783	STANLEY L. SHAW	30340	TIM LINK	37221	HEALTHCARE TELECOMM.	47906	MALCOLM ANDRISANI
22003	DON C. ECKHOLDT	30458	JAMES W. TAHLER	37601	LAILA HAMPTON	47966	DONICOLM AUKERMAN
22030	KENNETH P. HYNES	30467	J. HILL	37662	STANLEY W. STILL, CPA	48018	KEITH A. DEVORE
22032	DAVID FISCHER	30542	JAMES SCARBOROUGH	37830	JAMES B. BALL	48020	CONNIE SIEH
22039	ANDREW P. SAGE	30605	GEORGE A. HOUGH, III	37931	WARREN LAMBERT PHD	48020	ROGER LA POINTE
22041	JAMES L. SKILTON	30909	BRUCE A. WIGHT	37931	M. CARL BECKER	48034	FRED APEL
22094	JOHN P. RETELLE	31099	R. & S ACCOUNTING & TAX SERV.	38024	S.P. WELBORN, JR.	48043	HENRY W. MILLER II
22101	DR. E.L. JOHNSON	31201	PAUL E. LYLES	38111	FRANKLIN JORDAN	48050	ROBERT E. LAURION
22124	WARREN B. SAUDERS	31501	JAMES C. JOHNSON, ACCOUNTANT	38186	AMERICAN CONTRACT BRIDGE LEAGUE	48058	MICHIGAN REPRO-GRAPHICS
22124	J.S. SHIMP	31510	LORENZO E. HAND	38655	OXFORD SOFTWARE CO.	48063	THOMAS B. LIGHTBODY
22151	DR. STEVEN A. WARNER	31707	BEV. B. HARRIS	39204	BOBBY G. YATES, VICKERS, INC.	48063	GEROGE J. MACK
22170	OLEG V. FEDOROFF	31909	VINSON & ASSO.	39301	PET FAIR/HOBBY FAIR	48067	FRANZ H. BREIDENICH
22180	DANIEL D. ARDEN	32073	GENE W. ASCHENBECK	39301	J.B. GRIFFITH	48075	JIM MITCHELL
22191	J.C. CASEY	32205	JOHN H. SAARI	39530	J. WESLEY TOCHE, III	48077	RICHARD M. KIRKA
22193	ROBERT L. PORTER	32221	RICHARD HARRELSON	40118	IAN G. ELLIS	48079	LEONARD HOOL
22201	ALLAN ARMUS	32241	J.B. JAFFA	40299	JAMES R. SCHMIDT	48084	MR. MICHAEL B. ACHORN
22301	WOLFHARD RAMM	32308	BEN JOHNSON ASSOCIATES INC.	40383	WAYNE D. THOMPSON	48085	KEN ABBOTT
22306	EUGENE C. BOUNDS	32405	JAN CRANE	40506	DAVID ALLEN	48089	ROBERT CLYNE
22901	COLIN T. RAE, SR.	32405	BOBBY R. LUDLUM	41653	ERNEST BROOKHART	48103	PETER W. MEEK
22903	S.N. VINES	32541	DENNIS A. BYRNE	42301	J.A. BRYANT	48104	DR. JAMES L. TAYLOR
22903	GLEN L. BULL	32605	JOHN C. SODERSTUM	42701	OFFICE ASSISTANCE, INC.	48105	MEL L. BARCLAY, MD
23113	JESSE BAILEY	32608	CARLOS DURANDO	43081	MARVIN A. KATZ	48125	RAYMOND R. PASK
23150	RIX N. DOBBS	32714	W. PORTER WIGHTMAN	43085	GEORGE W. CALLENDINE, JR., PH.D	48161	BRENT LOCKSMITH
23454	WILLIAM R. MARRIN	32725	ROBERT F. GELLERMAN	43205	CHILD ABUSE PROGRAM	48197	JOHN HUGHES
23462	KATHERINE PITSLIDES	32746	CHARLES N. RITTER	43209	DECOR CORP.	48197	CLIFFORD HARRISON
24017	DR. STUART P. JACKSON	32748	BILL GALBREATH	43212	EMEN KING-SMITH	48202	S.W. JAMES
24073	JIM OVERFELT	32771	GEORGE CHENEY	43224	EDWARD BERANEK	48203	HERMAN TANN
24078	ROGER A. HOUGH	32803	STEPHEN E. DOLIVE	43537	FRANK T. ALLEN	48227	MICHAEL D. LOFTUS
24153	DR. HOWARD D. SACKETT	32901	ROGER POLLOCK	43611	JIM FRY	48412	ADVANCED ELECTRONICS INC.
24503	WILLIAM H. PRESSLY, JR.	32937	BERYL SMITH	44106	DR. MICHAEL E. MIGUIRE	48446	DR. RONALD L. LYMING
25303	DR. HENRY O. COLOMB, JR.	32952	ARTHUR J. PHILLIS	44124	GARY L. WHEELER	48453	BLUE WATER HOMES, INC.
25325	R.C. MORRISON	32961	BUD L. HOLMAN	44256	ROBERT W. BROWN	48706	THOMAS J. BAILEY, CPA
26719	JAMES A. WHITMAN	33023	DAVID MORRIS	44260	W.E. CLAXTON	48731	RONALD L.F. IRIS
27104	JERRY WHELAN	33023	AL O'HARA	44278	DON MARTIN	48906	DAN HANSEN
27106	HAROLD MENCK	33064	DAID - HOLSTBIN	44280	D.L. RINGWALT, JR.	49007	ERIC SCHREUR
27253	JACK L. PICKARD	33138	DOUG INGRAM	44505	K.C. KUNIN, M.D.	49012	B.A. THUNMAN, WBISG
27405	TERRY HOUGH	33143	W. WALDO LYNCH	44646	VERIL E. TURSKY	49017	STEVEN HARKE
27510	MARION BARKER & PAUL COLBY	33144	YGNACIO MORENO	44720	CHARLES MARKINSON	49038	BOB MARTIN
27511	JERRY W. RICHARDSON	33156	ALPHA BUSINESS COMPUTERS	44805	JOHN ROWSEY	49269	ROGER KLOEPPER
27695	DENNIS HERMAN	33157	JOSEPH FERREIRA	44815	NORMAN E. MURRAY, ACCOUNTANT	49301	IVAN BELYEA
28207	JOE D. CLARK, CPA	33165	R.S. KULZICK	45202	EUGENE M. ROTHCHILD	49454	RAY DE MEESTER
28403	ROBERT SANDERS	33177	WILLIAM KLINE	45227	WM. E. JOHNSON	49505	BUSINESS GUIDANCE SERV., INC.
28532	JOHN KIEFFER JR.	33311	BENITA BRANDT	45231	STAN MUSE	49862	PETER JONAS
28739	C.D. GIBBS	33528	ALBERT B. MEWHINNEY	45242	J.S. LOWREY	51334	MERLE JOHNSON
29205	RALPH A. RUSCETTA	33606	TONY LETO	45342	JERRY ALTHOFF	51501	STEPHEN DAVIS
29260	JOHN E. MC MURRAY	33715	CAPT. DOUGLAS W. HILLER	45433	BRIAN MAASS	52240	JAMES FLUCK
29405	T. KAPRA AMSTAR CORP.	33715	ARTHUR B. CHAUSMER	45433	MARTIN RICHARDSON	52240	GERALD STAMP
29407	GARY LUNSFORD	33731	JOHN W. WAECHTER	46112	STOREN COMPUTER SYSTEMS	52761	WALTER A. THROM
29445	STANLEY YOAKUM	33860	WILLIAM W. STEWART	46168	TERRY E. PLANK	52806	RICK AHLGREN
29578	GEORGE D. SACKETT	33881	MILTON P. CHARTER	46176	TIM DEATON	53115	ARTHUR L. JOHNSON
29681	HOOKS ELECTRONICS	33940	KENNETH G. HADCOCK	46208	J.B. ORRIS	53115	ARTHUR L. JOHNSON
29715	JOE AVAMPATO	33940	DAVID FOGELGREN	46222	PAUL W. HERRING	53154	CHRISTIAN S. SZOT
29730	BILLY MCFADDEN	33952	E.H. BROWN	46229	ELDON HAWKINS	53186	G. SUTTER
29928	PATRICK D. KEENAN	35803	DOUG ELGIN	46236	R.L. WOLF	53186	JEROME J. MONFRE
30001	JOE FOPPANO	35807	DR. ROBERT A. BROWN	46307	GEORGE M. POLLINGUE	53211	MICHAEL B. SHEFFEY, CPA
30067	DR. R.E. CUNNINGHAM	36104	WILLIAM L. IRVIN	46320	THIEL COHEN	53219	RICHARD J. KAERER
30083	JOHN R. PEDERSON	36109	CHARLES A. HIGHTOWER	46321	JOHN B. MALLOY	53222	KENNETH J. HINDL
30117	E. PERRY WALDREP JR.	36111	JAMES C. MATTHEWS	46506	ORTON MILLS	53226	ARVIS A. KRAETSCH
30117	RICHARD T. JONES	36582	ROY KEELEY	46590	VILAS E. DEANE	53558	STAN SITTS
30201	THOMAS CAGE	36590	MICHAEL R. WILLIAMS	46615	DENNIS HEAD	53705	ROGER L. MOORE
30213	BEN M. AMOSS	36604	DAVID PATE MD	46703	JIM SHEARER	53706	GLENN SATHER
30306	ROBERT H. WALLING	37133	WILLIAM Q. CRICHLAW	46733	JIM HOWNSTINE	53713	PAUL BENDER
30309	PAUL E. YERGENS	37212	STEPHEN D. KEEL CPA	46807	DANIEL E. BERNING	53715	TOM SCHEIBL
30322	DARRYL NEILL	37214	S.H. PEARSALL	46815	DON SLANE	54552	TOM SCHEIBL

54901	COMP CARE COMPUTER SYSTEM	640/B	DON CAMPBELL	77005	BILL ARNOLD	83340	JOHN HERBERT
54914	RONALD SAMSON	65202	D. J. MANSON	77013	GEORGE HODGES	83401	DAVE SEVY
55108	DAVID DUGAN	65656	STEPHEN C. BABBIT	77058	MARY C. FERGUSON	83402	DON STAPLES
55113	BRUCE M. WEBER	66085	DARREL SCOTT	77074	APPLIED METEOROLOGY, INC	83701	J. R. KILLIAN
55407	CHRISTOPHER A. SIMS	66208	MORTON JACOBS M.D.	77081	J. T. MCCORMACK INC.	83704	JOHN D. MUTCH
55413	EUGENE PISHKO	66614	GLENN BECKER	77227	W. D. HAMMOND	84057	DELTA DATA SYSTEMS, INC.
55417	MICHAEL CANNY	66839	SPENCER SMITH	77277	D. W. HUGES	84058	ERIC N. SKOUSEN
55423	VICTOR H. HEINER	67114	O. K. HUDSON	77373	ROBERT D. PEARSON	84117	KENNETH G. MEYER
55438	DARRELL S. BRAUN	67203	DENNIS KELSEY	77450	JAMES TATE	84119	W.L. PENROSE
55438	G.M. CZAJKOWSKI	67203	GEORGE W. LANDIS	77511	ROGER SHULKIN	84123	DAVID R. ROWLEY JR.
55441	GERALD B. SHEBLE	67203	JOHN CAMPBELL	77539	BOB ALEXANDER	84319	EUGENE K. ISRAELEN
56241	STEWART JESERITZ	67401	JOHN P. CAMPBELL	77571	JAMES G. HALES	84321	HIYOSHI YOSHIMOTO
56601	RANDY WALKER, INC.	67601	CHARLES VOTAW	77571	SIMPLE OPERATING SYSTEMS	84335	NEAL LANGERMAN
58108	JIM HOFF	68133	JERRY FRANCIS	77843	A.G. BLACKWELL	84401	NORMAN WORTH
58274	JOSEPH C. TATE	68154	KIRK BENEDICT	78022	JAMES H. EARLE	84403	STEVEN E. KAMMEYER, M.D.
5855	JAMES A. ENGELTER	68510	A. GERKEN	78102	BOB MILLER	84404	PHILIP W. PRATT
59601	OSCAR M. CARLSON	70001	LOUIS J. ULMER	78102	OAK LAND COMMUNICATIONS	84532	CLAUS & ADKISON CPA'S
59624	CUSTOM COMPUTING	70032	GARY J. TRICHE	78209	PEPOS S. DOUNSON	84604	WENDELL ALLRED
60040	ARNOLD B. TONI	70065	MIKE PICKARD	78216	DOM SCHULMAN	85004	PERRY S. MINAS
60045	ARTHUR SCHMIDT	70068	GARY BORNSTEIN	78216	NEIL A. MORGAN	85013	MARGARET ZINKY
60053	ROBERT SCHUMAN, CPA	70112	TULANE UNIVERSITY	78284	DR. BENNETT DYKE	85014	METTEE/MCGILL/MURPHY
60062	NORMAN BLOCH	70112	ELLIOTT M. BAIN, CPA	78539	RONALD C. KETTERING	85015	DONALD H. SMITH
60067	A. L. WUDI	70122	JOHN S. SIMONTON	78705	DOROTHY L. DAYLEY	85023	L.W. (VERNE) DISNEY
60068	ED COUDAL	70394	LAWRENCE J. MILLER	78759	WALLACE TUTEN	85034	R. L. SANDERS
60068	JAMES M. ULLMAN	70422	ROBERT A. LANDRY, JR.	78934	JAMES H. WITCOMB	85037	MARTIN J. MEYER
60090	MARTY LEIDER	70601	CONWAY STONE MAGEE, M.D.	79007	WILL GRAHAM	85224	WAYNE R. ANDERSON
60093	THOMAS A. WICK	70663	JOHN BAILEY	79106	EARL MACK HUNTER	85251	RICHARD ERIBES
60115	PRO. WALLACE R. MCALLISTER	70663	C. T. BOMBECK M.D	79606	NIEL NORWOOD	85258	FLYING BUFFALO CCS. INC.
60302	ALAN ROBINSON	70806	WILLIAM J. WEMPEN	79709	RICHARD GERTH	85281	JACK R. RUSSELL
60430	ROBERT M. JOHNSON	70815	E. FRANK COP, CPA	79925	ANDREW LUNA	85282	GREG CLARK
60439	WILLIAM S. POINDESTER III	71037	WILLIAM R. PERRY	80005	JERRY RUNYAN	85301	FRED KARTH
60515	PETER BOCCUZZI	71037	O. D. LA FARGUE, JR	80010	JACK BETTIS	85308	ROBERT J. ALEKSA
60540	KENT HELLER	72143	DON WEIR	80026	WOLFRAN KASENITR	85602	LARRY C. SCOTT
60608	PHYLIS McDONALD	72653	ROBERT FLOYD	80030	LARRY COSTA	85613	HAROLD ADAMS
60614	BRUCE B. BOYD, M.D.	72745	JEFF BANKS	80044	RICK DOWNS	85703	LOYD HOPPER
60616	CHEM. DEPT. DE PAUL UNIV	73527	LARRY E. SMITH	80110	PAUL WHITAKER	85708	ARTHUR E. GARTNER
60616	TATE YOSHIDA	74003	WAYNE MCGINNIS	80122	CHARLES R. BARNES	85710	DUANE MILLER
60652	EDWARD CRASS	74012	DONALD WORTMAN	80210	BRIAN PHILLIPS	85710	JAN BUGLEWICZ
60659	DANIEL D. STUHLMAN	74012	FRANK LAUGHLIN	80211	JOYCE N. STIVERS	85710	HILBERT SCHOUTEN, CPA
61604	DOCTORS&HOSPITAL CREDIT OFFICE	74012	DALE TRUMBLE	80302	J.A. VULETICH	85712	SAUL LEVY
61701	M.R. CONRAN	74104	DAVID J. KEENER	80302	PETR BECKMAN	85716	D.M. HUNTER
61801	HENRY MURRY	74561	PERRY GARST	80303	RICHARD ROOK	85719	ALLEN COLEMAN
61912	JIM WIMAN	75001	DAVID W. NOELL	80307	DONALD LEWIS	85732	SOUTHWEST IMPEX
62025	E.R. WILLIAMS	75007	WILLIAM E. YAGER	80309	WILLIAM D. MCCA J.R.	86001	B.W. DAVIS
62901	RICHARD M. JOHNSON	75042	RICHARD L. SOLOMO	80501	JIM SWARTZ	86301	CLAUDE A. NEWMAN
63011	GARY MANN	75088	CONRAD ROMBERG	80521	GORDON WANG	87108	V.E. DUDLEY
63011	DENNIS L. DOEFLER	75090	STEVE MARUM	80525	JERRY HUMMEL	87110	RICHARD FLORES
63017	LOREN C. FORRESTER	75149	STEVE LEAKE	80631	JOSEPH F. HAEFELL	87112	MAJOR WILLIAM R. MUSSATTO
63031	CHRISTOPHER R. PUSCZEK	75229	JAMES W. TODD	80759	COMPUTERSMITHS	87502	LAIRD GRAESER
63031	KENNETH O. HENDERSON	75240	GERALD E. TRENT	80906	ROBERT G. MACDONALD	87544	ROBERT R. SHOWALTER
63032	GENERAL SYSTEMS, INC.	75243	WALTER B. RICE	80907	DEAN HENDRICKSON	88003	CHARLOTTE MCCARTHY
63119	GEORGE A. REID	75711	DAN STEPHENSON	80907	FLEMING ELECTRONICS	88003	JOHN A. LUDWIG, BIOLOGY DEPT.
63121	J. WM. MUELLER	76011	JAMES ANNIN	81212	VERNON ESTES	88003	M.H. BERNSTEIN
63121	CARL E. SCHAFFNER	76011	MIKE STUTE	81301	MEL MATIS	8800327VICENTE D. VILLA	
63128	ERIC D. NULSEN	76043	G.D. EGGER II	82071	RON JOHNSON, WASHAKIE CENTR.	88004	BLUEMONT CONSULTANTS, INC.
63141	SHERWIN MALT	76105	MICHAEL J. RIGGS	82401	JOHN W. DONNELL	88201	WARREN EHLERT
63146	SIDWELL SOFTWARE, INC.	76118	HARVEY F. DESOTO	82426	BRUCE W. BREGSTROM, CPA	89004	JOHN SON
63366	L.R. SLATTERY	76302	JOSEPH E. HAYS	82801	T.J. DAHLQUIST	89101	A.G. ZENGER
63401	WM. REMILLONG	76437	JOE J. COOPER	82801	GLENN DORSCH	89104	NICK LEONIS
63801	LEE A. BOWMAN	76801	PAT RUDESEAL	82834	JAMES S. GUYTON	89107	GILBERT FIRMINICH
64055	DON WEEKLEY	77002	A. RAY-DEWITT&CO., INC.	83338	CHARLES H. CORRELL	89121	FRANK A. WHITAKER JR.

89511	J. E. PEPPE	92077	JEFF WELLS	94040	STEVE GRIMES	94039	DAVID DAVISON
90045	E.E. ST. JOHN	92083	M.G. CORDERO	94040	KEN SMITH	94541	KIM ELLISON
90045	HUBERT H. LOVE JR.	92103	JOHN F. GERGURICH	94040	GORDON MACBETH	94542	S.Y. TANG
90045	LELAND C. SWENSON	92121	ENERGY SCIENCE LAB., INC.	94044	ROBERT LEE	94542	JOEL H. FINK
90045	BEN CURTIS	92342	JACK RENBARGER	94063	HERRIMACK SYSTEMS	94544	MIKE JORY CREATIVE SERVICES
90048	JOSE L. RODRIGUEZ, CPA	92344	DONALD D. FAWCETT	94063	E. NOWAK	94546	ROBERT MCCLOUD
90066	SAM EL-HAI	92373	WILLIAM E. JAVERTT	94066	CARL R. KOENIG JR.	94547	SANFORD S. LAVINE
90066	WILLIAM PROUD	92376	DORR STUART	94066	NANCY J. MOHLER	94549	RALEIGH ELLISEN
90066	ALFRED J. MONROE	92507	WILLIAM H. ORTTUNG	94070	GEORGE HEBLACK	94550	GEORGE BUSH
90068	EDWARD P. ANCONA, JR.	92627	JOHN T. LEWIS	94086	PAUL KOHLMILLER	94550	ELMOND HOLBROOK
90213	JOHN RILEY	92627	JAMES E. FRANKLIN	94086	AMTEL SYSTEMS CORP.	94553	EARL S. HALE
90230	PHILIP MASSIE	92635	MARSHAL T. BRECHT	94086	WALLACE E. LIN	94556	LEONARD MORGENSTERN
90241	BRIAN HOLMES	92646	RICHARD T. MCCARTNEY	94086	GEOFFREY ROSE	94558	ERNEST LEE ABBOTT
90247	JOHN ARMSTRONG	92647	STEPHEN SHARON	94087	MIKE AHLMANN	94558	NORMAN DELEVZE
90247	GEORGE POND	92649	MARL RUBIN	94087	HUGH FROBACH	94558	FRANK THOMPSON
90247	RONALD MASAOKA	92653	M.G. LOWE & COMP.	94087	JIM MELLENGER	94558	GREG W KELLER ACCOUNTANCY CORP
90260	PAUL W. PHISTER, JR.	92660	DICK LEWIS	94087	JOE BYERLY	94558	PHILIP RUSIN
90265	CHARLES E. LOVE	92670	PHILIP D. COREY	94087	GENE F. WALTERS	94558	ARCHIE E. RIVETT
90266	G. R. TELLE	92675	DUNCAN MORRILL	94102	LINDA HARRINGTON	94560	ROY LATHAM
90272	JANETTE RAINWATER	92690	WILLIAM B. MCDONOUGH	94102	FRED D. LONSDALE	94563	FRANKLIN LEW
90272	E.F. GRANT	92707	J.F. SULLIVAN	94107	PATRICIA SALBER	94563	JOSEPH STRABOLA
90274	CLINTON LEW	92714	ROBERT MACINTYRE	94107	DENNIS J. LUNSFORD	94564	ALFRED DEL SIMONE
90274	S.M. CLEMENTE	92714	CHRISTOPHER A. PESAVENTO	94109	JACK EASON CONSULTANTS	94566	HEXCEL CORPORATION
90274	GEORGE UEBLE	92804	WALTER JELUM	94109	RAYMOND S. CRANDELL	94568	DOUGLAS J. SEMON
90280	EDWIN YARGA	92804	B.A. WILKINSON	94110	ART SIEGEL	94572	GERALD GROW
90291	JAMES MC GINTY	92805	LESTER WELCH	94110	PAT MALONEY	94572	FRANK ERNST
90402	J.V. HOLDAM	92806	FUMIHIDE NAKAMURA	94110	GUY YAGUROFF	94572	DAVID B. WOOLARD
90404	GHOLOMOSSEIN M. BIGDELI	93004	STEVE NOLL	94111	VICTOR BARKHORDARIAS	94574	JAMES THOMPSON
90504	LEROY KERANEN	93010	F. L. ABERNATHY	94111	WILLIAM PARKER	94577	ED ELLEFSEN
90505	KEN BOYCE	93030	ROBERT J. GLASSNER	94114	WILLIAM PARKER	94578	KENNETH LETSCH
90630	CRAIG BURNE	93030	JOHN M. OLSON	94117	T. GOR. BALDWIN	94578	RICHARD K. GARY
90701	RALPH CARTER	93105	TONY DENNING	94118	DR. ROGER P. FRIEDENTHAL	94580	CHARLES MCDOWELL
90701	COMPUTER OPTIONS	93110	ED LISOTA	94118	JERED S. NELSON	94583	LOUIS E. ALMGREN
90717	JOHN MADDOCK, INC.	93302	WEBSTER ELECTRONIC CO.	94120	WILLIAM HARMON	94583	WAYNE HATAYAMA
90717	JOHN J. CLEARY	93308	ROBERT FLAMING	94122	WALLACE FRIESEN	94585	GEORGE RIDDL
90745	DONALD NIERAETH	93308	JIM LEEK	94122	BOB BARLOVE	94585	ROBERT GARY
90807	R.H. DAVIS	93401	JEFFREY W. FISHER	94123	CLYDE STEINER	94585	JOHN H BARCLAY
90814	DON E. APPELEY	93403	SUSAN I. KNOWLES	94127	TED JOE	94589	YVETTE HOFFER
91001	WILLIE L. PATTERSON	93422	JIM FITZGERALD	94131	WILLAM WRAY	94595	JODY PONCE
91040	BOB KUEHER	93436	RICK GROSSMAN	94133	TOM MC CARTY	94596	SARAH WASSERMANN
91103	THE CHANDLER SCHOOL	93436	LIBRARY-COMPUTER SCI.CORP	94134	JOHN T. ARNOTT	94596	HAROLD JEFFREY
91105	ROBIN WATKINS	93455	FUEL INJECTION	94134	VERNA M. WALLACE	94598	FREEMAN E. GRAY
91106	TONY B. ANDERSON	93510	BERT BERRY	94303	EUGENE DONG, M.D.	94598	JEFFREY STRAUS
91107	W.F. PFEIFFER	93534	TRINDEL J. FERGUSON/K-COMP	94304	KEN KELLEHER	94598	MICHAEL REED
91202	MINGHU WU	93534	JAMES DUNN	94306	GLENN C. STEINER	94598	JOHN SEIDELL
91321	W.T. HICKS	93550	C.J. REYNOLDS	94404	E.W. AMES	94601	THOMAS J. CROTHERS
91335	TODD ZERVAS	93555	DIXIE D. GREEN	94501	BETH SPRINGSTON	94602	GARY L NOCK
91344	J.W. CURTIS	93555	CARL H. MORLEY	94508	RANDY FISCHER	94602	DAVID RISARD
91344	ROBERT C. RADFORD	93555	G.D. THIGPEN	94509	BOB BEAVER	94606	CALIFORNIA APPRAISAL SERV.
91351	G & R MARRIQUE	93662	TORII MARKET	94509	JAY M. HUBERT	94608	VIROLAB, INC.
91364	JOSEPH KEMP	93711	JOHN M. KIRBY	94518	JAMES LIND	94609	BRUCE KLICKSTEIN
91401	ARTHUR HORNER	93727	DOUG VAUGHN	94518	LARRY ANUTA	94609	BRETT JOHNSON
91401	LARRY W. RISSE	93901	CHARLES BAILIN	94518	INDUSTRIAL GRAPHIC ARTS	94611	THOS SUMNER
91403	GILBERT J. GILBERT	93923	STEPHEN B. KAHN	94518	LARRY T. NICHOLSON	94611	M.H. CHENEY
91413	DAIL A. DEVILLENEUVE	94002	HERMAN THOMS	94518	JAMES L. ADAMS	94611	ANN GORDON MC STAY
91601	CARVEL STEPHENSON	94010	DR. D.W. CARLSON	94519	JOHN FARNHAM	94619	MARTIN RISARD
91604	DAVID H. MONKARSL	94015	KURT H. HECKSCHER	94523	ANN HERNANDEZ	94703	DANIEL BERKEY
91705	DENNIS M. SHARPE, PRES.	94022	J.S. STUJZINSKI	94523	COLLEGE PARK HIGH SCHOOL	94704	RICHARD M. BETTS
91750	JOHN J. MC NULTY, JR.	94022	J.N. CHRISTIANSEN	94523	KEVIN NEWMAN	94704	STEVEN W. BARWICK
92007	EDWARD BOSS	94022	MR. ED HERNANDEZ	94523	BIRNEY ADAMS	94705	ROBERT MILLS
92008	GARY MOSSELLE	94022	J.C. SPOUTON, ELECTRA MOD	94526	WILLIAM SHIELD	94710	BOB SIMMONS
92020	DONALD R. HENRY, D.C.	94022	L.K. KOEHLER	94530	FRANK DAVIDORE	94710	JOHN LORAM, V.P. ENGR.
92027	L.A. BEEL	94025	DONALD DUNN	94530	DICK WILLIAMSON	94720	ALAIN HENON
92037	HARRY G. BULESTIN, MD	94030	CHARLES MAYNARD	94533	ARMANDO PICCLOTTO	94803	MELODIE M. YAMADA
92061	EDWARD L. HOPKINS	94030	DATA METHODS & SERVICE CO	94539	DONALD R. GOTHOLD	94804	DOUG & KATHRYN SISKIN
92065	DANIEL CROSS	94030	RICHARD D. HIGHSMITHS			94805	HUGH POWER

94806	JAMES M. KELLEHER	95826	LUCIEN A. GRONDIN	98502	TIM LINEHAN
94806	WALTER E. LAUGHLIN	95826	SACRAMENTO COUNTY APCD	98506	RICHARD BOEDEKER
94901	DAVID KARP	95831	GARY KAWAYE	98520	DON A. KEEFER
94901	WAYS & MEANS, INC.	95831	ALAN PRITCHARD	98532	MIKE PETRA
94903	PAUL MCGOVERN	95926	S. JOSEPH TOY	98532	CHEHALIS TRUCK & DIESEL INC.
94904	JIM FURMAN, FURMAN SOUND, INC.	95926	MARC REMERO	98661	GREGORY HAZEN
94925	G.M. SPITALNY	95927	MARC CULLEN	98665	THOMAS G. PARSONS
94925	M. DOONAO	96328	STEVEN TOWNSEND	98840	JOE THOMASON
94947	STEVE BROWN	96701	LLOYD D. STERLING	98902	JAMES EVERT
94957	LHARY MEYER	96720	EUGENE PETERSON	99328	CHARLES H. CHEEVER
94960	HENRY SPELLMAN	96778	W. B. HOWARD	99336	R. CHARLES SMITH
95003	WILLIAM PROUDFOOT	96813	WALTER YOUNG	99345	SANDPIPER FARMS INC.
95014	ANTHONY R. BECCIA	96817	PHILLIP M.K. LAU	99350	JAMES WILLARD
95030	TOM MC DANIEL	96819	ALAN KOBAYASHI	99362	TUCKER LAND CO.
95050	COATS WHEEL BALANCER,CORP	96822	ANDREW M. LUM	99501	DAVE DUKE
95051	DOUGLAS W. TOPHAM	97005	RONALD K. MELOTT, PE	99504	JOSEPH MAGUIRE
95062	PETER WILDE, KINNEDIC LABS	97030	CHARLOTTE L. MONTGOMERY	99559	BASIC BUSINESS SERVICES
95065	SAL BASILE	97042	REX BREUNSBACH	99559	GALEN DIRKSEN
95070	EUGENE E. JONES	97110	GENE WHITLOCK	99615	JOE STEVENS
95112	GEORGE EWERS	97113	PATRICK M. CASTLE		
95117	RALPH GETSLA	97115	RON MILLER		
95120	CHARLEY LICHTENSTEIN	97116	ROBERT A. BROWNING		
95120	PHILIP E. BAILY, M.D.	97123	JOE QUAGLIANI		
95120	S M SCHREIBER PROCOM	97130	RONALD G. LARSON, PE		
95121	DANIEL BROWN	97204	FRANK WALL		
95123	DAVID TOMLINSON	97214	JACK C. RILEY		
95129	GARY CYR	97216	R. EDWARD COOK MD PC		
95129	RONALD J. CANTONI	97223	DAVID PARKER		
95130	R.J. RALDIRLS	97225	C.K. SHANKS & ASSOCIATES		
95133	ROBERT JEHLE	97266	MR. JAMES E. MCCASTON		
95133	RUSSELL D. OESER	97331	MICHAEL SCHUYLER		
95140	WAYNE EARTHMAN	97339	JOE SNYDER		
95153	JOHN DIBBLE	97403	JOEL W. MC CLURE		
95157	DENNIS ALLEN	97601	ENGINEERING DEPT.		
95205	COMPLEAT DATA SERV.	97702	ROBBIE LAMBERT		
95209	ALAN REYBURN	97827	KENT HUG		
95223	SHEILA STERLING	97914	JOHN FOWLER		
95247	LEIGH ROBINSON	98004	BOB ANDERSON		
95340	DAVID D. BROADWATER	98005	J. FRANCIS SMITH NRCC		
95354	T.J. COYLE	98006	JAMES A. BLOOMFIELD		
95380	JOHN E. BAKER, M.D.	98008	KENNETH E. MATTHEWS JR		
95401	DON JACKSON	98008	WARREN KIMBER		
95401	ROBERT LONG	98011	JOE S. CREAGER		
95402	C.R. BRISCOE	98024	N.W. GRONLUND		
95404	ROGER T. SMITH	98027	W.D. DRUMMOND		
95437	JEREMIAH MURPHY	98040	JOEL E. HAAS, M.D.		
95440	J.J. KENNEDY	98055	DON MURPHY, PTL COMPUTING		
95444	ANDREW BOND	98104	ROBERT SIGLEY		
95451	STOKES LADDERS, INC.	98112	JOHN CHENG		
95476	KENNETH S. MCTAGGART	98115	JOHN F. SCHNEIDER, PRESIDENT		
95540	ALBERT S. BURROWS	98116	BURKE C. DYKES		
95608	CAMERON CLARKE	98116	THOMAS C. LOOMIS		
95608	HERB HOLEMAN	98117	PAUL B. ONCLEY		
95608	MILT WEINBERG	98119	G.E HUGHES		
95616	RICH NUCCITELLI	98119	RICHARD B. KERN		
95620	STEVE O'NEILL	98121	KENNETH L. WALL & CO.		
95624	RICK MURRAY	98125	ROY GILLETTE		
95628	ALAN J. ANTOS	98145	PAUL S. PERSON		
95661	GEORGE ERIO	98195	MICHAEL GLASS		
95676	ROY CHADWICK	98199	WARREN V. REBACK		
95678	DICK WAGNER	98223	JOHN PRESLEY		
95688	HARVEY SHAPIRO	98371	WILLIAM LAWSON		
95730	REV. LEONARD N. SHAHEEN	98402	LARRY CLARK		
95814	GENNIS & ASSOC. DIV. NOLT	98406	ROBERT SCHAEFER		
95817	KEITH CRANDALL	98499	JIMMY GULLEKSON		
95821	HANK SHAPIRO	98502	TONY LERNER		

Disk Simulator Patches

INSUA:

In Vol. 4, No. 2 of The Compass, Joe Maguire reviewed the LS-100 Disk Simulator and offered a copy of his patched BIOS to anyone who sent him a disk and return postage. I contacted Joe, and received back copies of DRC's implementations on a 5.25" disk (DRC supplied an 8" disk which I could not read). The implementation is for Lifeboat CP/M 2.21.

I am running Lifeboat CP/M 2.22, and many addresses are different in that version. I have managed to get all three of DRC's programs up and running under CP/M 2.22 and make the same offer as Joe did, i.e. I will copy these programs for anyone who sends me a blank disk and return postage.

Charles Prohaska, RD #1, Box 234, Dingman's Ferry, PA 18328

Vendors Column

In this column we reprint vendors' descriptions of their own products. Of course, INSUA makes no claims about the quality or merchantability of any commercial software, hardware, or services. **Caveat emptor!**

MICRO SUPPORT

As of February 1, MICRO SUPPORT offers TELEPHONE HOTLINE SUPPORT for microcomputer end-users, from 9 a.m. to 5 p.m. (PST) weekdays.

MICRO SUPPORT's HOT LINE is unique in the marketplace as it is neither product specific nor requires bulk time purchase. There is a minimum charge of \$15 for any part of the first 15 minutes, \$1/minute thereafter, real time. Charges begin when you are actually on the line with a phone tech; if your question can't be answered, there are no charges at all. Payment is by Mastercard or Visa. All calls are received, entered, and time-monitored by a call log system. Serial numbers for software remakes (those MICRO SUPPORT is licensed for) are provided by publishers and manufacturers and available for instant identification by MICRO SUPPORT.

Should the end-user's problem require on-site support, phone techs will transfer the call at no charge to an account manager who will arrange for services by one of the 150 MICRO SUPPORT contract programmers throughout the Bay Area. Charges for on-site services are generally \$50/hour, though it can vary for especially esoteric or complex tasks. An elaborate database and search program allows rapid identification and location of appropriate programmers based on from two to fifty criteria per search.

MICRO SUPPORT is an unusual company in many ways. Offices are located on a 3000 sq. ft., 3-story houseboat in Sausalito, CA. The first deck houses the phone techs and account managers, communications systems with capacity for 24 in-coming lines, multi and single user computer systems and (only in California!) a sauna. The second deck, complete with kitchen and circular hanging fireplace, is used for conferences, programmer gatherings, and a general retreat from screens and pressure. Staff meetings take place daily over lunch at the conference table.

In its first year MICRO SUPPORT has grown tremendously, expanding both its client and programmer bases. This is probably due to the fact that the company focuses on the satisfaction of the client and the professional and personal needs of programmers and phone techs. There is a distinct "esprit de corps" among the programmers, who make themselves available to each other whenever needed. MICRO SUPPORT offers skill-specific contracts and opportunities to train on a variety of computers and both pre and post release software, both at MICRO SUPPORT offices and through alliances with OEM's, publishers, and companies such as the Whole Earth Software Review. Phone support specialists have the opportunity to "get out of the house" frequently to carry out on-site contracts, preventing burn-out and enhancing income.

Both clients and programmers interested in learning more about this organization are invited to call anytime and ask for Susan or Mike at (415) 331-5034.

\$ \$ \$

INSUA:

I have been a user of a North Star Horizon for five or six years, and of an Advantage during the past two years. I seldom run anything except APL and am authorized by TELECOMPUTE, Inc. of Canada to issue licenses in the USA for their TIS-APL to run on North Star computers.

At a reasonable cost to cover expenses I can furnish a demo disk with TIS-APL and several interactive learning programs thereon. After a trial period of 30 days the recipient may simply return the loaner disk or apply the initial cost toward a permanent APL license in his name. Cost of the demo disk with a TIS tutorial of some 49 pages is \$250.00. If the decision

is made to return the material, \$200 will be refunded. If the recipient decides to keep the material, the loaner disk may be exchanged for one with permanent license info thereon. The flip side of this QD disk will also contain some 45 utility programs useful in APL wordprocessing, file copy/other manipulations. Also, a complete TELECOMPUTE reference manual will be included, all for an additional \$250. The APL character set is software generated at the time of boot-up. Symbiotics can provide extended character set APL EPROMS for Epson MX-80 and MX-100 printers, and also an APL EPROM for the Heath H-19 terminal which provides a low cost APL terminal for the Horizon, for example.

Write: SYMBIOTICS
W. E. Claxton, proprietor
431 Mishler Road
Mogadore, OH 44260

\$ \$ \$

INSUA:

Today's computers are becoming an essential part of the household, as well as business, in this country. The advantages of using computers increases yearly, but an important factor we often forget is that computers are operated by people who risk straining, or even damaging, their eyes when watching a computer screen for prolonged periods.

To prevent this strain from occurring, our laboratories have developed the Eye Care CR-39 lens to protect the computer operator. This lens has been fitted to a light-weight frame designed specifically for the computer operator.

The Eye Care CR-39 computer glasses are now available in a variety of models at reasonable prices. Dealer inquiries invited.

Write: Pacific Trade
16 Emery Bay Drive
Emeryville, CA 94608
(415) 655-9340

\$ \$ \$

TurboDOS Made Easy

By Steven Marks

The complete users guide to the TurboDOS operating system

If you use or sell TurboDOS multiuser systems, there is one book to help you get the most out of your system. Whether you are a newcomer to computers or an old hand, TurboDOS Made Easy will increase your competence and the effectiveness of your system.

The author, Steven Marks, brings a rich background in teaching TurboDOS, and a gentle hand in explaining difficult to understand concepts, to TurboDOS Made Easy. Mr. Marks, both a computer expert and a psychologist, kept the novice user in mind, writing in an easy to understand, conversational style designed to get anyone started quickly. All TurboDOS commands and options are explained in context of actual applications to help everyone get the most from their systems.

The August 1984 Microsystems magazine article, "Getting The Most Out of TurboDOS" by Steven Marks offers a brief glimpse of the power this book opens up for users at all levels of experience. Chapters include the following:

- About Computers.
- Getting Started with TurboDOS.
- Logging on.
- CP/M-like Functions.
- Other TurboDOS Commands.
- The Master Processor.
- Printer Functions.
- The Monitor.
- Commonly Performed Functions.

Appendices:

- Error Messages and Explanations
- COPY Options
- CP/M Equivalents of TurboDOS Commands
- Index

198 pages, soft cover, \$24.95

Pro Comp Systems, Inc.
333 West 52nd Street
New York, NY 10019
(212) 246-0074

\$ \$ \$

Letters to the Editor

INSUA:

Some time ago I wrote concerning a request for information how to screen dump the graphics I can generate with GBASIC on to my Epson printer. It's been months since I wrote and in the meantime I figured out a sloppy but effective way of getting the job done. The procedure is as follows. Load drive 1 with NSDOS and GBASIC. Load drive 2 with the demo diskette that came with my Advantage. Using CF (copy file), copy "EPSON" from drive 2 to drive 1. After creating graphics on screen with GBASIC, get out of GBASIC by typing in BYE. At the # sign type in GO EPSON. Then type control-T. My printer is an Epson MX-100 with Grafrax plus. In order to get all of the needles to print I had to open the printer and reset the internal switches to get the 8th bit thru. The switch is on the Epson 8150 interface (S2 switch, function 4).

The problem with this approach is that all of the stuff on the screen including the BYE and #GO EPSON also appears on the page. If someone knows a simpler way of doing this, preferably by just using control-T at the end of run, I'd appreciate hearing about it.

Sincerely,
Fred B. Abeles
7001 Runny Court
Frederick, MD. 21701

INSUA:

Your advertised message about INSUA sounded like a dream come true, for I can find no North Star Users Group in this region, despite presence of over 100 North Star Advantages in departmental offices at the University of New Hampshire and another 50 or so in faculty ownership, including my own.

I use a North Star Advantage with Epson MX-100 with Grafrax, at home

virtually every day, mostly for writing books, magazine articles, notes and letters. The last 3 of my 54 published books were done on Wordstar; a tenth article for Scientific American also (accepted but not yet scheduled). I feel fairly glib with it, but find little use for the CalcStar, SuperSort, SpellStar that I purchased. Much more use for Documate for indexes, table of contents, etc. and other jobs of sorting alphabetically. I have a Hayes Smartmodem 300 baud for telephone connection to DEC-1090s at the University of New Hampshire where I time-share along with other professors and the students.

Articles I would like to see, compatible with the North Star Advantage, include:

1. Smooth scrolling under Enhanced WordStar (promised but no key pattern shown in the new manual)
2. A poor-man's program for the North Star Advantage (64k) comparable to Sci-Mate (Institute for Scientific Information, 3501 Market St., Philadelphia, PA 19104) for handling bibliographic information: index card format, search for key words, sorting alphabetically or by key topics. The University library here has the package for a North Star with hard disk, but cannot legally share it with faculty, especially for a smaller amount of RAM.
3. A program for retrieving 2x2 slides by subject, sortable, printable, perhaps by Accession Numbers if other marking is complex. Perhaps #2 would do that job also.

Cordially yours,
Lorus J. Milne, Professor of Zoology
1 Garden Lane
Durham, NH 03824

(We will shortly be running a favorable review of BIBLIOGRAPHY, distributed by Digital Marketing Corporation, 2363 Boulevard Circle, Walnut Creek, CA 94595 (800) 826-2222; try it, you may like it! -- Ed.)

INSUA:

I must correct part of the commentary of mine which appeared in Volume 4, No. 1. I never could figure out how a configuration byte of OFFH could cause my disk controller not to read the second side of the disk; it turns out that my disk controller was at the time on the verge of complete failure, and that was the source of the problem. Since I replaced the board a few weeks ago, I got curious and tried changing the configuration byte back to OFFH (instead of OC3H) using SYSGEN, and lo and behold, it still was able to read the second side of the disks!

I had been very puzzled about why DCOM seemed to perform in such a strange fashion; now I know the problem wasn't DCOM, but my controller board. Now I am very puzzled as to why changing the configuration byte seemed to make a difference - for a while - in the ability of a sick controller board to read the second side of a disk!

Very truly yours,

Robert L. Porter

INSUA:

First and foremost, congratulations on the appearance and content of the magazine. It is a truly professional publication and provides an on-going reference source that is invaluable to North Star users. May your Tribe increase!

Second, a plea: Has anyone out there come up with a method of converting existing North Word data files to Word Star (or other word processor) data files without re-inputting every character?

Reason: Since the beginning, as a confirmed North Star backer and buff, I have used the proprietary software made for North Star including NorthWord and InfoManager. We have a large number of tariff pages - over a thousand - on disk under North Word.

My secretary threatens to join the Foreign Legion, however, whenever any extensive changes to the commodity index are required, for they are produced in two 35-character columns on a page, and North Word has no columnar editing capability. Word Star, among others, does. My North

Star dealer, Brillig Data Systems (great name!) informs me that no revision of North Word is available or in prospect. I am therefore reduced to contemplating a change to some other word processing software that has columnar editing capability, such as Word Star's latest release. But the prospect of re-inputting around half a million words does not enthrall me.

Any ideas?

Sincerely, and gratefully,

Wm R. Harmon
Vice President
Matson Navigation Company
333 Market Street
PO Box 7452
San Francisco, CA 94120

(I have run into the same problem with dBASEII, and this is what I do. A dBASEII file has a non-ascii header, but all the information following the header is pure ascii. I find some way of saving the file in memory above 4000. Then I reboot in CP/M, invoke POWER! (or a similar utility), and inspect the material in memory. If I can locate the beginning of the ascii data material, I can save that to the disk. When I inspect the resulting file with WordStar, I find that it is nicely formatted in columns. I can edit the material with WordStar (using column commands if necessary), or re-absorb it into a dBASE file using the (filename).TXT sdf option. Perhaps some variant of this process will do the trick for you. Of course, you cannot perform the operation on any more material than will fit into memory at a given time, so it may be necessary to break your original material up into smaller files first. --Ed.)

INSUA:

Further to my comments on the undocumented commands in North Star BASIC: I have dis-assembled BASIC 2.1.1, and have found the following commands:

SIZE - at 258b hex
INFO - at 2623 hex

There are also three new error messages ... namely:

INDEFINITE VALUE - at 1527 hex
UNFORSEEN (something) - at 1531 hex
NO FILE MANAGER - at 153B hex

All of this assumes BASIC's Origin at 1000 hex.

I am unaware of the operation of the first two, and of the meaning of the second three! If documentation is not available from North Star, then perhaps you could publish this information, in the hope that a better hacker than myself and can? provide the answers!

I look forward to receiving the free disk, with Secretary and MODEM712, although you may be interested to know that this is the very first document I have typed with MAGIC TYPEWRITER - which is available in both North Star DOS (which I am using) and CP/M versions. It is very good, and has no special requirements other than a Z-80 (NOT 8080!). Particularly in view of the price! Just \$15.00 on disk, with 90 pages of documentation, spiral bound! I also should mention that it is not just a word processor, but a data-base manager as well! Not bad for \$15.00 I'd say.

MAGIC TYPEWRITER has proved, so far, to be extremely easy to learn, and is available from:

C.D.E. Software
2463 McCready
Los Angeles, CA
USA 90039

Thank you for taking the time to write. I look forward to receiving the COMPASS.

Yours, very truly

David S. Young
1926 Monteith St.
Victoria, B.C.
CANADA
V8R 5X5

INSUA:

Several things I have noted in the microcomputer market are causing me concern:

a) North Star disk format is less and less frequently listed in ads for 8 bit software.

b) Sales of 5.25" diskettes frequently state "soft sector only" or don't mention sectoring at all.

c) Most new languages or new compilers are being offered to run under PC-DOS or MS-DOS, or, to a lesser extent, CP/M-86.

What I would like is some way of adding the proper hardware and software to my system to result in an 8-bit/16-bit co-processor and a disk controller which could read and write some of the more popular 5.25" as well as 8" disk formats, in addition to the North Star format. I suppose this would also require some method of separating 8-bit and 16-bit memory on the bus.

For the next several years, I do not anticipate a need for a multiuser system, so the North Star Horizon upgrade boards do not seem to be the answer.

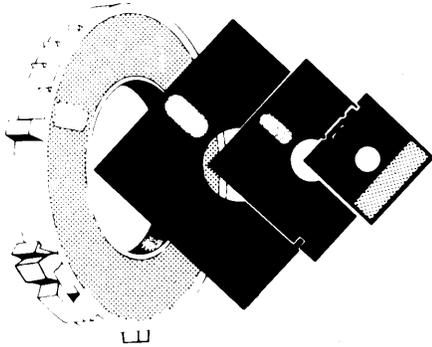
What I am after is to be able to continue use of all of my 8-bit software, be able to run at least MS-DOS, and be able to read and write enough other disk formats, in addition to North Star, so that I can be assured of being able to obtain useable commercial software, as well as blank diskettes, for the foreseeable future. Ideally I would also want to be able to exchange data between the two processors through internal memory transfers.

If this does not appear to be too tall an order, I would appreciate INSUA members' advice on how this might be achieved. While not by any means a competent digital circuitry technician, I am no stranger to a soldering iron, having assembled and debugged my original Poly 88 computer, my Heath terminal, and the two 32K static RAM boards in my Horizon.

Sincerely,
Oliver C. Stokes

(One probable future for the Horizon lies in single-board computers on S-100 boards which you can simply insert along with your other boards--these could be PC- or MS-DOS compatible boards which can be run as an option to your Z-80 board. The trick will be to make them share resources with your other software and hardware, so that you can, for example, send files from computer to computer. Any further ideas from other readers? --Ed.)

Want To Move Your Computer Files To Another Computer?



File transfer on
5 1/4" and 8" drives.
IBM, CP/M, Apple,
Northstar, Mag
Tape and 200 other
formats supported.

**Contact Lee T. at (619) 693-1697
Computer Conversions, Inc.**

11419 Menkar Place
San Diego, CA 92126

Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

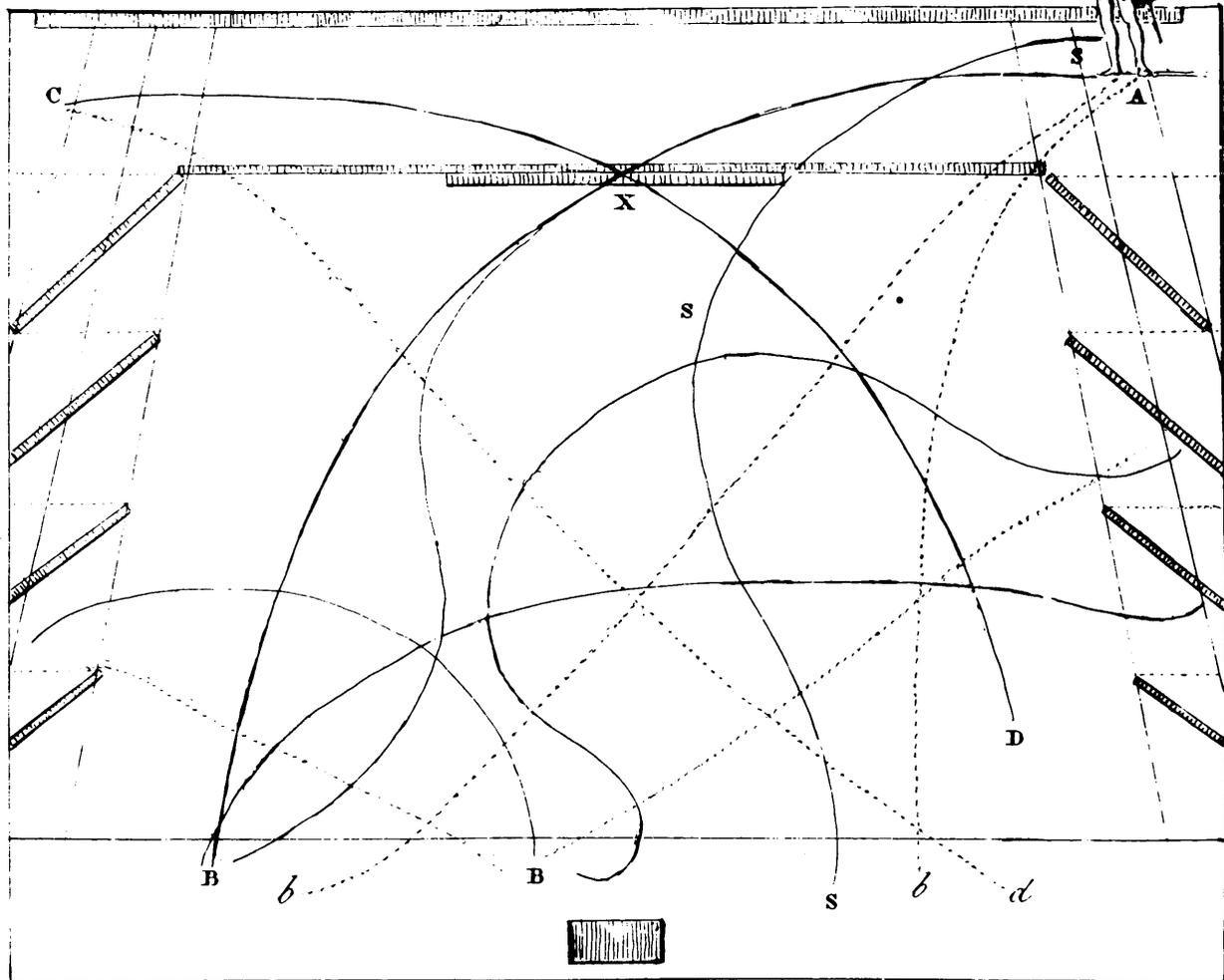
Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203

TATE 276H
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

The Compass

International NorthStar Users Association

Volume IV No. 5



Notes from the Editor

As the holiday season is upon us, the INSUA Board of Directors sends Greetings of the Season to all members, past, present, and future. We're 1,500 strong, and maintaining our size in spite of strong competition in the marketplace from other manufacturers, including You Know Who.

Speaking of You Know Who, discussions at the most recent Board Meeting, and letters and articles in this Compass and last, suggest that very high on the list of priorities for Horizon and Advantage users is the need to become disk-compatible with the rest of the world. This should not mean turning in North Star's for other computers, but finding ways to make North Star products read soft-sector disks or vice versa. (Programs are available on soft-sector machines to read one another's files, though of course many machines cannot run one another's programs.) We earnestly solicit comments on this topic, experiences with installing and running either hardware or software fixes, with matching up machines, and so forth.

Achieving disk compatibility is, of course, a job for experts, though we could all benefit once the experts get the solution(s) worked out. A problem which relative novices occasionally report with INSUA disks is that the disks **won't boot up**. We have occasionally had problems with defective disks, and are always willing to replace a disk found to be defective at no cost to the member. However, it is important to realize that with the exception of the disks which have contained updated versions of DOS, the disks we ship do not have operating systems on them. The reason for this is that the operating system is a piece of software which the user must purchase from the manufacturer, either North Star, or another. If we supplied disks with the CP/M operating system already on them, we would be breaking the law by giving away software which the manufacturer of CP/M sells--we would be engaged in software piracy!

So, if you acquire a disk from INSUA, chances are that you will have to boot up first with your own operating system, whether DOS or CP/M; then insert your INSUA disk in drive A or B, read the directory, make a copy, or do whatever is necessary. Remember, though, that you must boot up first with your own operating system!

Similarly, some members have been confused by flippies, as opposed to regular floppies. Flippies can be identified by the fact that they have **two** square write-protect notches on the edges rather than one. Use the disk the regular way for one side, and flip it over for the other side. If you are not certain what side belongs with which operating system, try both. A CP/M side should show CP/M files when read from DOS, but you will recognize immediately that it's not a regular DOS disk. If you use CP/M to **dir** a DOS disk or a DOS side of a floppy, you will get a message telling you that there are no files on the disk. Don't believe CP/M, at least not right away. Try reading the same side with your DOS operating system first, using **LI**.

If none of this works, it is possible that your disk is truly defective--in that case, send it back to us.

The Editor of Compass is working on issue # 6, and can use more copy, as always! Please send disks to the INSUA address. We can now read double-sided disks as well as single-sided, so we're keeping pace!

The Compass

The Compass is published every two months by INSUA, the International North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, Compass, P.O. Box 2910, Fairfield, CA 94533.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

Contents

- 2 Edgar F. Coudal **The Trailing Edge**
Reminiscences: 6 years as a N* user
- 3 John A. Bryant **38,400 Baud: Hardware**
Higher baud rates for your terminal
- 4 **Membership List**
Canadian Members of INSUA
- 5 John A. Bryant **38,400 Baud: Software**
Higher baud rates for your terminal
- 8 Alan H. Nelson **BIBLIOGRAPHY: A Review**
For databases with variable fields
- 11 **Membership List**
"International" Members of INSUA
- 12 Pepos S. Dounson **Commentary**
More on SECRETARY
- 13 Alan H. Nelson **Sound Steward: A Review**
User's experience installing fan filter
- 16 Warren Lambert **Lazy Man's Secret Handshake**
Using handshake routines already in OS
- 18 Jim Lind **N*SORT with BASIC 5.5**
Routines to make N*SORT work with 5.5
- 20 Warren Saunders **Commentary**
On Phantom Signal and Fan Filters
- 20 Terry Hough **Here's the Meat**
Listings left out of last issue
- 22 Ted Carnevale **Soft-Sector on the Horizon**
Making the Morrow DJ/DMA work with Horizons
- 24 Charles A. Hightower **Computer Books**
Reviews of four books on Z-80
- 26 **Vendors Column**
The Computer Journal, Typesetting
- 27 Joe Maguire **Dear Joe**
Letters to inquiring readers
- 29 **Letters to the Editor**

The Trailing Edge

By Edgar F. Coudal
(Copyright November, 1984)

(This column was originally written for InfoWorld's Contributor's column, and is reprinted here with the permission of the author.)

"The guys on the leading edge of technology get icicles in their beards, seagulls in the face and arrows in the back. When you're on the trailing edge, you're shielded from the turbulence, it's very warm and there are a lot of other people in the same fix, eager to help."

So said a speaker at a User Group meeting of a leading super-mini manufacturer.

It made me feel better.

In my public relations, advertising, and free-lance writing business, I use a North Star Horizon micro built in 1978 (serial # 00012, the oldest such working beast known to North Star). Since our clients all are in high-tech electronic businesses including computers, I occasionally feel guilty about my Z80-based machine with its 58K of usable RAM, its no-longer supported North Star Disk Operating System, the incredibly dumb Soroc 120 terminal, and the 18cpm Diablo 1610 printer.

Every day, I deal with Gould super-minis and CompuPro MP10s and laser drives and other state-of-the-art hardware. I write about integrated software and spreadsheets and syntactical checkers and Butterfly-Hart sorts and document indexers and outliners and UNIXes and Adas and Modula-2s and so on.

But there I am, discussing those up-to-moment wonders and marching to the beat of drummer who laid down the bass line six years ago.

Yet my word processing software (LetterGo and WordStar) handles my most fevered output easily, probably yawning to itself all the while.

My homebrew mailing list and financial packages (written in BASIC!) go about their business efficiently and without complaint.

The communications software--a Pleistocene version of Christensen's Modem7 running on a secondhand PMMI board--uploads and downloads and

unerringly connects me to The Source and Knowledge Index and all the great BBS's out there.

But best of all, there are the other North Star users. Because the company always has been cavalier about end-user support, North Star owners banded together to form the International North Star Users Association, with more than 1,000 members and an invaluable "quarterly" magazine that comes out occasionally. The Chicago area subgroup interacts frequently and freely.

Questions are asked and workable solutions supplied, or at least suggested. Hardware and software modification help is always available from someone who has had the same problem. Examples: A Chicago user helped install my 5-meg hard disk. A San Francisco wizard left the six-line Hex code I needed to activate my RAMdisk spooler in a message on the PROXIMA (North Star only) BBS in San Leandro, CA.

So here I am, back on the Trailing Edge of microcomputer technology, but doing everything I want to do with speed and efficiency. That speaker was **correct**. It is warm back here.

But he forgot to mention something.

Even on the Concorde, the Trailing Edge gets there only a moment or so after the leading edge...

#

Edgar F. Coudal is President of BC Communications, Inc., and Coudal & Associates, Ltd., two Chicago-based marketing communications firms.

38,400 Baud: Hardware

By John A. Bryant
6648 N. Canandaigua Rd.
Holcomb, NY 14469

Modifying the Horizon for 38,000 Baud From the Serial Port

Some of today's video terminals (such as the Wyse 50) are capable of running at 38,400 baud, much faster than the baud rates available from the standard Horizon.

The two Horizon serial ports have adjustable baud rates, but changing baud rates must be done by hardware changes involving strapping a baud rate header located on the motherboard. Despite the fact that even in the early days of microcomputing (which is when the Horizon was designed) video terminals were available that were capable of running at 19,200 baud, the highest baud rate obtainable without modification is 9,600 baud.

Two Different Methods

I am aware of two different modifications that can be made to the Horizon that will yield a baud rate of 19,200 baud. The good news is that if these two modifications are combined, you can have 38,400 baud, along with all the other standard baud rates down to 150. However, you cannot have both 150 and 300 baud available at the same time.

I've stacked these changes and have been pleased with the results of operating my Wyse 50 terminal at 38,400 baud. I'll state the two changes separately so that you can step up to 19,200 baud by applying either procedure, or to 38,400 baud by applying both.

The first modification described is taken from Compass, Vol. I, no. 2, and the second from the October, 1984 issue of Polaris, published by the North Star Computer Society in Seattle. Bob Stek, the author of the article in Compass, said that the procedure was sent to him by Dr. Jim Byram of Massachusetts, while Jeremy Shapiro, who described the procedure in Polaris, said he got his information from Harry Binswanger of the New York North Star Users Group. Combining the methods was my own idea.

The first modification only works with Z-80 processor boards using an 8 MHz crystal, which is the usual configuration. Whether you have an 8 MHz crystal can be determined by looking at the crystal at location 1F on the Z-80 processor board to see if "8.0000" is stamped on the metal can. If you don't have an 8 MHz crystal, you'll have to use the second procedure and content yourself with 19,200 baud.

If all you want is 19,200 baud, I recommend using the second procedure since it does not require tying the Z-80 board to the motherboard by a soldered-in wire, and it does not change the other baud rates available at the baud rate header.

A Word to the Wise

CAVEAT: Mucking around inside your expensive computer with a soldering iron is not something you should do unless you have an appropriate iron, appropriate solder, and appropriate know-how. If you aren't confident of your skills, I'd advise you to obtain the assistance of someone with the requisite knowledge and experience. DON'T try this with a "soldering gun" or the old antique your grandfather used to mend tin roofs, or even with the soldering iron you used to build that Heathkit stuff back in the '60s, and don't use the wrong kind of solder.

A Neat Tip

This is not a course in soldering technique, but I will repeat a tip described in the Compass article. Any time (such as here) that a modification requires soldering onto an IC pin or bending up an IC pin, you can avoid messing with the IC itself and can make the procedure easily reversible by using this tip. Obtain an IC socket and an ordinary header, both with the same number of pins as the IC involved. Solder

the socket to the top of the header, pin by pin, but don't solder any pins that are going to be bent up; instead, bend up the socket's pin. Instead of soldering to the IC pin, solder to the socket's pin. Then plug the IC into the new socket, and plug this 3-layer sandwich into the socket that the IC originally went into. To reverse the modification, pull out the sandwich, unplug the IC, then plug it back into its original socket. Neat.

The Two Methods

METHOD ONE

- 1) Locate the three equidistant plated-through holes that are just below IC 6G on the Z-80 processor board (viewed from the component side of the board), and solder a length of insulated #22 or so wire to the rightmost hole (which is just below the space between pins 7 and 8 of IC 6G).
- 2) Pull out IC 7C from the motherboard and bend up pin 11.
- 3) Solder the other end of the wire to pin 11, then plug everything back in (leaving pin 11 bent out).

What you've done is taken pin 11 of IC 7C off of the 2 MHz bus clock and put it onto a 4 MHz signal you've tapped from the Z-80 board. The effect is to double all the baud rates on the baud rate header.

METHOD TWO

- 1) Connect pin 1 to pin 14 on the special clock header at location 1D on the motherboard.
- 2) Connect pin 2 of that header to pin 12 of IC 7D on the motherboard. (Don't bend this pin up.)
- 3) Wire the baud rate header (location 2D of the motherboard) so that pins 3 and 4 are connected to pin 1 instead of wherever they were previously connected.

This method (by itself) doesn't double the other baud rates available at the baud rate header; it merely puts 19,200 baud on pin 1.

Final Adjustments

Whichever method you choose, or if you use both methods so as to yield 38,400 baud for the video terminal, don't forget to change the baud rate settings of the devices plugged into the Horizon (or restrap the header in the Horizon as appropriate). Then check your work over, turn on the computer, and enjoy the speeded up terminal response.

I would suggest you pencil the changes that have been made into the schematic in the back of the Horizon Computer Manual, and reflect the baud rate changes in the chart on page 75 of the Manual.

#

CANADA NAME LIST

In accordance with INSUA policy, we list here current Canadian members of INSUA in zipcode order, with only "zipcodes" and names. The purpose of this listing is to assist Canadian INSUA members in getting in touch with one another by consulting telephone books. The privacy of members who have chosen not to have their names listed in local telephone books should remain protected.

CANADA

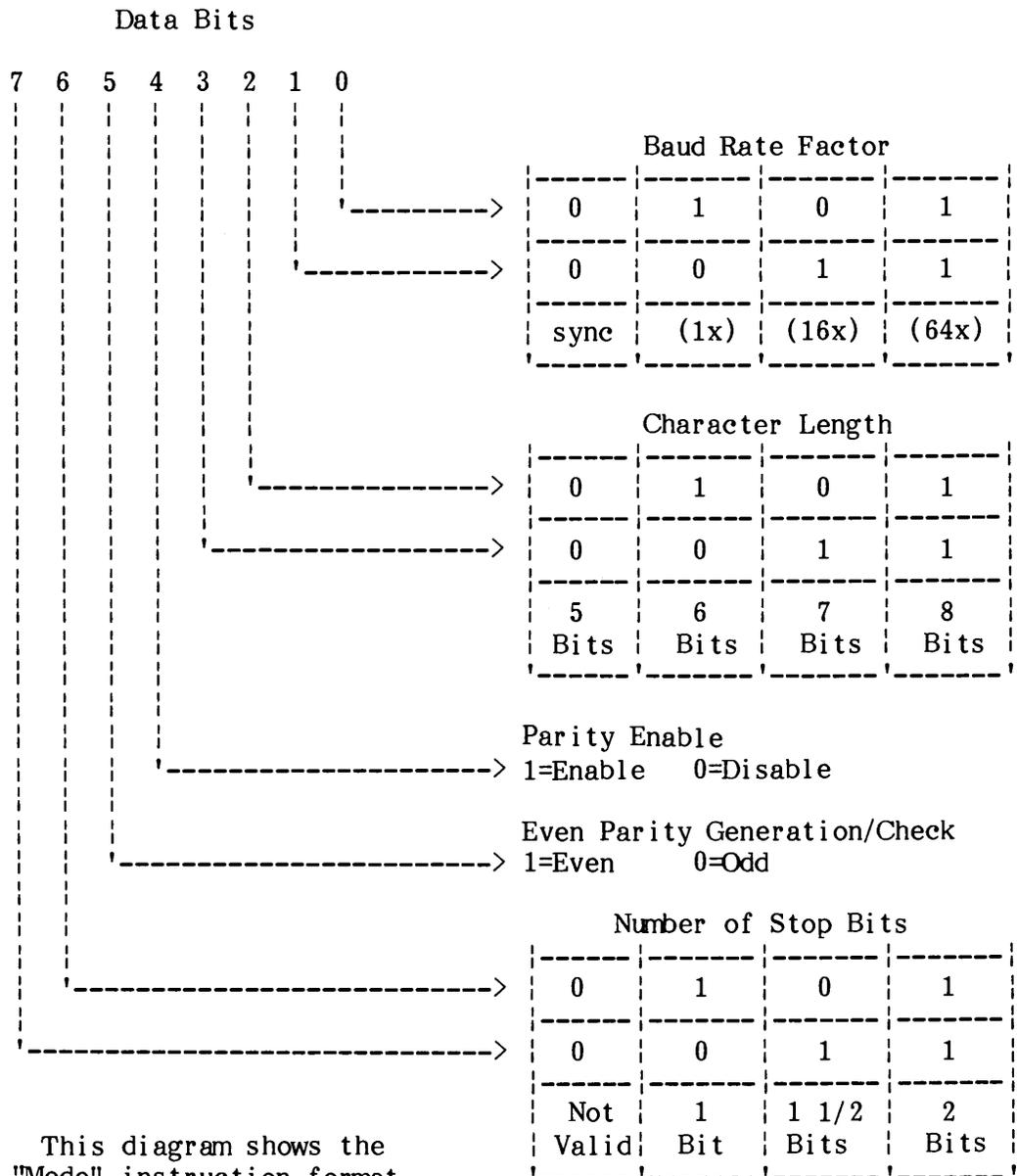
Doug Appleton Regina, Sasck. S4V 1K9	Michael O'Byrne Ottawa, Ontario K2C 1Y2
De Jager Dowsview, Ontario M3J 1P6	Ronald Hayter Vancouver, B.C. V6J 1S5
Bill Darker Orillia, Ontario L3V 6H3	Bob Dawson The Pas, Manitoba R9A 1R8
William Scott Seaforth, Ontario N0K 1W0	The Flyer People Vancouver, B.C. V6B 3X9
Hiroto Saka Edmonton, Alberta T6J 3P1	T.M. Guest Houston, B.C. V0J 1Z0
Richard Derksen Winnipeg, Manitoba R3N 0B3	Peter J. Gauthier Moncton, N.B. Canada E?C 8H7

38,400 Baud: Software

John A. Bryant
6648 N. Canandaigua Rd.
Holcomb, NY 14469

8251A USART Instruction Format for Asynchronous Mode

(Wherein the reader is also told how to drive the serial ports at 38,400 baud and faster without hardware changes)



This diagram shows the "Mode" instruction format.

See below for the "Command" instruction format.

(Stop bits only effect Tx; Rx never requires more than one stop bit.)

The North Star Horizon uses 8251 USARTs (Universal Synchronous/Asynchronous Receiver/Transmitter) on its serial ports, but nowhere in any of the manuals have I been able to find a chart explaining how to set the two serial ports for baud rate, stop bits, character length, and parity by means of instructions to the 8251s. A clue appears at Page H-3 of the "Getting Started" portion of the System Software Manual, which is where the I/O routines for the DOS are set forth. At location 29A3, where the machine language instruction does a Move Immediate of 0CEH into the A register (which value is then sent to the USARTs for both the first and second serial ports), the comment says "2 STOPS, 16xCLOCK, 8 BITS, NO PARITY". However, there is no explanation of how the value 0CEH was determined, or what values to use to change any of those parameters. The same instruction and remark are found on page 11 of the System Software Manual Addendum dated 7-11-80, for memory location 0A91H.

The above chart is from an 8251 data sheet, and shows how to program the 8251s in the North Star for other word lengths, clock speeds, parity, and stop bits.

To use the chart, first decide the configuration wanted. For example, if you want to run at a baud rate factor of 64x, use 8 bit words, 2 stop bits, and even parity, the instruction to the 8251s would be 1111 1111 in binary, which translates to 0FFH. Note that 64x will be 4 times SLOWER than the 16x that North Star DOS is originally set for. (The 1x mode is 16 times FASTER.)

The new value should then be substituted for 0CEH at the appropriate location(s) in the DOS. Then when DOS is activated and goes through its initialization routine, the 8251s will be programmed for the new characteristics.

Changing the USART Mode Instruction for Higher Baud Rates

Although as set up the North Star Horizon and North Star DOS will give a maximum of only 9,600 baud at either serial port, higher baud rates are available by changing the instruction to the 8251. You can thus change the instruction to the USART for the left serial port and thereby run your terminal at 19,200 or 38,400 baud, without hardware changes on the motherboard (other than appropriate strapping of the baud rate header). By using a baud rate factor of 1x, you can get 38,400 baud from what would otherwise be the 2400 baud setting, and 19,200 from what otherwise would be the 1200 baud setting on that header. Note that that header is ahead of the USARTs, so running the left USART with a 1x factor does not affect the baud rates on the right port if you leave that USART at 16x.

If you want to make this change for the left serial port, use the DOS command LF to load DOS somewhere out of the way (eg, at 4000H), then load a Monitor program and change the appropriate memory location to the new value. Next, return to DOS and save the new DOS back to disk. Finally, rewire the header for the appropriate baud rate, bearing in mind the speedup or slowdown achieved by the new baud rate factor.

Here is a specific example (applicable to Release 5.2 or later) of how to change the left serial port so that you can run the screen at 19,200 or 38,400 baud:

Note: Characters you type are underlined; $\frac{1}{2}$ CR $\frac{1}{2}$ is a carriage return.

```
+LF DOS 4000 <CR>
+GO MONITOR <CR> (Use a version of Monitor that lies out of the
way of the area where you have placed the DOS.)
>DS 4892 <CR>
>4892 CE CD <CR> (If the Monitor program didn't display '4892
CE', then something is wrong. Don't continue.)
>OS <CR>
+SF DOS 4000 <CR>
+
```

At this point, and before pressing reset, you will need to rewire the baud rate header at 2D on the motherboard and set the baud rate on the terminal itself. For

38,400 baud, connect together 3, 4, and 14 on header 2D; for 19,200 baud, connect 3, 4, and 13. Then press reset or turn the computer on and boot up the modified DOS and your terminal should be running at the new speed.

You'll need to copy this modified DOS onto whatever diskettes you use to boot from. Moreover, if you also use CP/M, you'll need to modify CP/M to send the same baud rate factor to the left USART. Examine the source listing of the appropriate part of the System portion of CP/M to find the area to be modified. On my version of North Star CP/M, the source is found in a file called 'USER.'

Related Matters

Changing the baud rate by software has the disadvantage of requiring all your operating system software to contain the modification, or else you will have to keep changing the baud rate of the terminal. This disadvantage can be overcome if you achieve your higher baud rates by hardware changes rather than software changes. For hardware methods of achieving 19,200 and 38,400 baud, see the article in this issue of The COMPASS entitled 38,400 BAUD.

The "Command" Instruction

The "Command" instruction format for the 8251 is as follows:

- Bit 0 TRANSMIT ENABLE
 1=enable 0=disable
- Bit 1 DATA TERMINAL READY
 "high" will force DTR/ output to zero.
- Bit 2 RECEIVE ENABLE
 1=enable 0=disable
- Bit 3 SEND BREAK CHARACTER
 1 forces TxD "low"; 0 is normal operation.
- Bit 4 ERROR RESET
 1=reset error flags PE, OE, FE
- Bit 5 REQUEST TO SEND
 "high" will force RTS/ output to zero
- Bit 6 INTERNAL RESET
 "high" returns 8251A to Mode Instruction Format
- Bit 7 ENTER HUNT MODE (Has no effect in asynchronous mode)
 1=enable search for sync characters

North Star DOS sends the USARTs the command 37H, which is 0011 0111 in binary. This is found at memory locations 29ABH and 29AFH on page H-3 in the "Getting Started" portion of the Software Manual, and at locations 0A99H and 0A9DH at page 11 of the Addendum. The comment to the code at that point is "CMD: RTS, ER, RXF, DTR, TXEN", indicating that the high bits were set for Request To Send, Error Reset, Receive Enable, Data Terminal Ready, and Transmit Enable.

#

BIBLIOGRAPHY: A Review

By Alan H. Nelson

BIBLIOGRAPHY is a text-oriented program from PRO/TEM Software Company, marketed by Digital Marketing of Walnut Creek, CA. It can be used by itself in conjunction with many word-processing programs, but is also designed to work in conjunction with FOOTNOTE, a program which has been around for several years, and with NOTEBOOK, a program which for technical reasons is not available for the North Star Horizon, though it may work with other North Star Products.

What is BIBLIOGRAPHY?

The basic idea behind this program is that you maintain a file similar to a card catalogue in a library, with structured information such as author, title, date, and comments. This file can then be put to at least three uses:

- 1) You can create a working Bibliography for any given piece of writing, say an article or a book.
- 2) You can copy selected information from the library file **into** any other file, say an article or a book (for example into the footnotes).
- 3) You can print out a formatted version of the library file, either the entire file, or selected items from the file.

The real advantage of the process is that once you get the information for a given book or other item correct, you need never worry about getting it correct again in other places. BIBLIOGRAPHY will make perfect copies of the perfected original every time.

The %Key:

The first key to BIBLIOGRAPHY is the way its records are structured. Each separate kind of information is tagged in a special way. These tags help to identify the information which should go into any given field, but only the information (and not the tag) is printed out or transferred into the target files.

Several of the tags have a special function. The first is the Key tag, which contains the unique key by which the entire record is identified (and sorted against other records). The tag itself is identified by a % at the beginning of a line followed by the word "Key" followed by a colon, thus:

%Key:

The user inserts a keyword of his choice, followed by arbitrary tags of his own devising, followed by an optional field tagged

%Annot:

followed by a "%" all by itself on a line. Here is an example of a BIBLIOGRAPHY skeleton:

%Key:

%Author:

%Title:

%City:

%Date:

%Annot:

%

The tags are created and the "blanks" following the tags are filled in by the user, using a word-processing program.

The Library File

The library file consists of the skeleton repeated over and over again, with the relevant information supplied for each successive item. Since this function is performed easily by any competent word-processing program, the construction of the library file is not handled by any part of BIBLIOGRAPHY.

One function expected of a sophisticated bibliography program is to prompt the user for information. BIBLIOGRAPHY does not do this. The user who wants this function could write a simple BASIC program for the purpose. Alternatively, it is a simple matter to create a separate file consisting of just

the skeleton; this can be copied repeatedly onto the end of the library file (e.g. in WordStar with the ctrl-KR command). Another function expected of a sophisticated bibliography program is to locate selected items. Once again, this is not done by BIBLIOGRAPHY, but instead with the "search" function of the wordprocessor.

The one direct effect BIBLIOGRAPHY can have on a library file is to sort it, which it will do according to the ascii alpha-numeric value of the key fields. This function is performed by a separate program called "SORTBIB.COM."

The Text File

The text file is also produced by a wordprocessor. Exact equivalents of the key fields in the library file are inserted into the text file wherever a citation is needed. Since BIBLIOGRAPHY will eventually treat these citations in a special way, it has to be able to recognize them as different from the rest of the text. BIBLIOGRAPHY calls for the key to be preceded by a "%" and followed by any one of four punctuation marks, including "%)" and "%)".

Wherever BIBLIOGRAPHY finds one of these key fields in the text file, it will perform one or more of four functions, depending on the option requested when BIBLIOGRAPHY first comes up:

- 1) It will replace the key with the full citation, formatted normally (concatenated)
- 2) It will replace the key with a numerical reference to a resulting bibliography
- 3) It will construct a separate bibliography file, selecting from the library file only those items which have been cited in the text file
- 4) It will remove the key-word citation from the file, along with the special marks of punctuation

The Bibliography File

BIBLIOGRAPHY will create a bibliography file in any one of several formats:

1) It will create a concatenated bibliography, with all the information from each record run together into a single line

2) It will create a non-concatenated bibliography, with fields on different lines, as in the original library file

More Options

BIBLIOGRAPHY will include the annotation, or exclude it, according as the option list is set; it will put the author's last name in all caps or keep it as typed into the library file.

Listing the Keys

A program called LISTKEYS will list out all the keys for a given library file for on-screen inspection or for printing. Adding "%" as an additional argument to the LISTKEYS command will result in a file in which the keys alone appear, each preceded and followed by a "%"; this file in turn can be used as a dummy text file to force the creation of a file with the complete listing of the bibliography; alternatively, the dummy file can be whittled down with a wordprocessor to yield a formatted bibliography of selected items.

Ease of Use

BIBLIOGRAPHY is fairly easy to use but difficult to comprehend in all its possible ramifications. The number of options is large, and the number of permutations among all the possible options is even larger, so it is a chore to comprehend all of the "what if's", just as it is a chore to read through the documentation (and must have been a chore to write the documentation as well).

Nevertheless, if the user keeps in mind the limited applications required in any one session, BIBLIOGRAPHY can be understood. This is one of those programs which takes some effort, but will save enough time on a difficult task that the effort will be well rewarded. In actual use, it will probably be necessary to change only one or two options at any one session, and the rest can be used as originally set.

The most obvious uses of BIBLIO-

GRAPHY are two:

1) Insert citations into a text file. To do this, simply type in the key words (with code punctuation), and run BIBLIOGRAPHY (telling it not to generate a bibliography file. BIBLIOGRAPHY will ask for the library file (the source of the data to be inserted) and the text file (the target of the insertion). BIBLIOGRAPHY does the rest, except that the text file needs to be inspected again with the wordprocessing program, since many line-lengths will have been distorted because new material has been inserted in place of the short key words.

2) Make a bibliography, or a formatted selection from the full contents of the library file. To do this, insert the key words where they apply in the text file; request that the text file not be changed, but that a bibliography be produced. Again, the resulting bibliography file must be worked over with a word-processor, in particular to reformat the lengthy single-line citations into manageable paragraphs.

Other functions performed regularly include sorting the library, and printing out the keywords for quick reference. Your word-processing program can be used to append new items, delete, correct, or update old items, and to print out complete copies of the library file.

How useful is it?

BIBLIOGRAPHY works splendidly for its designated purposes. It is also useful for purposes it was not expressly designed for. Thus, for example, I have used it for data bases other than books. Because it pays no serious attention to the tags themselves other than the key field tag and the optional "Annot" tag in the closing field, it is possible to put information, text, or notations of any sort into a library file. A good example is a series of boiler-plate paragraphs, each with a different key. A letter can be made up with a salutation line followed by three or four keys, followed by a signature field. Run the letter through BIBLIOGRAPHY, and voila, the boiler plate is inserted in place of the keywords. The imagination alone will limit such applications.

Bugs?

BIBLIOGRAPHY is not perfect. A particular annoyance is its refusal to sort in any but ascii order. A text-processing program should, at a minimum, be able to fold to upper case; as it is, the last name ten Brink ends up following Zyzygy because small "t" has a higher ascii code than large "Z". Also, it sorts by keyword, not by last name. Some ingenuity is required to invent key words which will always give the equivalent of a last-name/first-name alphabetic sort. Finally (among the bugs I've found), it seems unnecessarily round-about to have to produce a key list from the library file first in order to produce a bibliography of the library file--why not a bibliography straight from the library file?

BIBLIOGRAPHY also requires a surprising number of files on disk. In addition to BIB.COM is SORTLIB.COM, MERGE.COM (to merge bibliographies) and LISTKEYS.COM; also OPTIONS.BIB, a file of less than a dozen bytes which is automatically changed as different options are selected. Finally, the disk must be able to hold the library file, the text file, and the bibliography file. In addition, it is useful to be able to keep your wordprocessing program available on drive A or another drive. In some cases the files can be allocated to more than one disk drive, but space can still be at a premium in any arrangement that requires so many files.

Companion Programs

BIBLIOGRAPHY is a perfect companion to FOOTNOTE, favorably reviewed in Compass, Vol. 2, no. 1, p 17. Brief key citations in notes will be filled in with BIBLIOGRAPHY before FOOTNOTE is used to reformat the file, placing the footnotes at the bottom of the page, or in a separate file. (Of course, a session with the word-processor must intervene to reformat paragraphs, including the footnotes themselves.)

PRO/TEM has also produced NOTEBOOK, a highly sophisticated database program which is designed to handle variable-length fields, in particular text rather than numerical information. Not so versatile as a fixed-field program

like dBASEII, it is superior in its ability to accommodate itself to fields whose lengths vary enormously, or whose maximum possible length cannot be predicted in advance. It is excellent for compiling a catalogue of book titles, which can range from one word to hundreds; it can also take annotations of indefinite length. If you're thinking that it acts something like the automated manager of a BIBLIOGRAPHY library file, you've got the point precisely.

NOTEBOOK, unfortunately, requires a real, undiminished, unencumbered 64K of memory, which no Horizon has (Randy Fisher can free up 62.7K, but no more). I mean to send the author of these programs a copy of this review with a plea that he produce a version of NOTEBOOK (perhaps a cut-down version) for his old North Star friends.

Documentation

Documentation is accurate and complete, but a little turgid, and inherently difficult since BIBLIOGRAPHY has so many options and permutations. Still, it is not ultimately difficult to understand. For myself, I would make one suggestion to prospective BIBLIOGRAPHY users, which is to create "prompting skeletons" whose tags are all the same length, for example:

%Key:
%Aut:
%Ttl:
%Cty:
%Dte:
%Annot:
%

This way it is possible to jump from one field to the other on the empty skeleton using, for instance, ctrl-X or ctrl-E in WordStar. With a skeleton of variable-length tags, it is necessary to use two or more keystrokes to move from a shorter tag to a longer one. ("Annot" must always be five characters long.)

I also recommend inventiveness in thinking up new applications for BIBLIOGRAPHY, which can be as useful for many other applications as it is for books.

\$ \$ \$

Available from:
Digital Marketing Corporation
2363 Boulevard Circle
Walnut Creek, CA 94595
(800) 826-2222

#

INTERNATIONAL NAME LIST

AUSTRALIA

Dr. B.G. Burfield
Adelaide, South Australia 5000

David Yates
St. Lucia, Queensland 4067

Bob Willson
Adelaide, South Australia

Armstrong & Partners RTA:LM
Toronto New Australia 2283

John Dent
Moonee Ponds, Vic 3039

Tectron Industries PTY LTD
Adelaide 5000

Trever L. Hill
Glossop 5343 South Australia

School of Applied Science,
Caberra College
Belconnen ACT 2616

D. & J. Gallagher
Mt. Pleasant, N.S.W. 2519

BELGIUM

Dr. De Ceulaer G.J.B.
B-2530 Boechut
Belgium

WANTED TO BUY

NORTH STAR 16K RAM BOARD FOR
HORIZON II. MUST BE IN WORKING
ORDER.

Veryl Turskey
2834 Poplar Ave. NW
Massillon, OH 44646

Commentary

By Pepos S. Dounson
445 West Sunset Road
San Antonio, TEXAS 78209

The last issue of The Compass with its articles on SECRETARY brought back visions and memories I made special efforts to forget.

MARYELLN was the first version of SECRETARY I ever bought. Later, I bought and am still using a DOS version of SECRETARY in which most, if not all, of the bugs have been removed by me. The only one remaining is the BLOAD and BSAVE bug, which I have not pursued.

My earlier version of SECRETARY would write over itself when the file I was SAVEing or NSAVEing was of a certain size or over. The problem is that SECRETARY keeps track of the number of blocks in the text. When it goes into the NSAVE or SAVE routine, the DOS interprets this figure as **SECTORS** (NOT BLOCKS). My version, with memory from 2000H to DFFFH, could only take 134 blocks or 67 SECTORS. When I told it to SAVE or NSAVE the text and it had in its booking accounting 100 blocks, it started saving 100 **SECTORS** which is 200 blocks. I didn't have 200 blocks of memory. It did just as it was told and when it got into the section that didn't have any memory as well as the PROM section, I got just what I expected ... partial destruction.

When I located the problem, I put in a routine to divide the blocks by two and then went into the NSAVE or SAVE routine. It now works fine. The space I used was in the message area where it said "PUSH RETURN TO CONTINUE". The message is shorter now.

When I wrote about my bugs the reply was "we haven't had that problem before". I see now, versions later, the problem still exists. One thing for sure, they sure were consistent, if nothing else! This particular problem would not appear in the single density version because there blocks and sectors are the same, but are TWICE the difference in double density.

I became aware of the saving problem when I attempted to NSAVE a 110 block brief I had been working on all day for

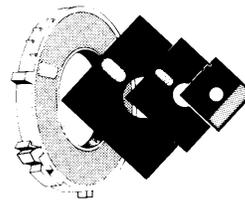
filing the next day, the last day for filing! I was in the elementary learning days of assembly language and spent quite a few months of night work analyzing and studying SECRETARY to find the problem. I even changed the name from SECRETARY to WORDFLEX to prolong my endurance because the name SECRETARY had now become a dirty name to me.

I pass this information on to INSUA users because the program is now usable for word processing with much greater pleasure with the bugs removed. Also, I changed my version so when the program is saved, it does not save so much extra space with it. It used to save four additional sectors instead of the customary two sectors. BASIC only saves two so why not SECRETARY? It left considerably more space for use. I kept wondering why a disk of SECRETARY text occupied so much space. Months later, I found out.

I hope this information will be of use to the users who will "cleanup" this buggy program!

#

Want To Move Your Computer Files To Another Computer?



File transfer on
5 1/4" and 8" drives.
IBM, CP/M, Apple,
Northstar, Mag
Tape and 200 other
formats supported.

Contact Lee T. at (619) 693-1697
Computer Conversions, Inc.

11419 Menkar Place
San Diego, CA 92126

Sound Steward: A Review

By Alan H. Nelson

Steve Hogan published an article on fan filters for the Horizon in Compass, Vol. IV, no. 1. Since I have been intrigued for some time about the question or problem of the fan on the Horizon, I acquired a kit and installed the filter. Am I happy I did it? Yes. Was it simple? Read on.

One of the first things I noticed about the Horizon I bought four years ago is that the fan seemed to have been installed **backward**. I wasn't very exercised at the time about whether air should be blown into the box or pulled out of the box, as long as the air was moving. On the outside of the fan, however, was a plastic air filter holder--no filter, mind you, just the holder. But since the air was being blown **out** of the box, the only function of a filter, if I did choose to install one, was to capture dirt that either came into the box from the outside, or was being generated inside the Horizon box. I didn't purchase the computer to filter the air in my room, but to compute; so I decided I wouldn't lose any more sleep over installing a filter.

I did of course realize that if I turned the fan around, I could put a filter on the outside, and thus trap any air-borne dust and dirt. Should I do this? If I did, perhaps things would go bloeey, or at very least the warranty would be invalidated. So I didn't.

Over the 48 months that my Horizon has been churning away, I noticed every time I opened the box that it was getting dirtier and dirtier. In fact, it was getting filthy. Air drawn in through the various holes and slots at a relatively high speed deposited its load of dust when it lost speed over the power supply, the chips, the wires, the S-100 card connectors: and in fact things got so bad that I couldn't read the numbers on the chips without wiping away the dirt first.

The odd thing was that my Horizon never seemed to notice. I have never had a minute's down time associated with any part of the computer other than the disk drives, which I have had aligned once. In fact, I just purchased a second-hand Horizon which worked like a charm, and

when I opened it up to change a header, I was appalled by the condition of the inside. Every corner of the box was filled with dirt strings, just like cob-webs. I haven't had time to do more than blow away the worst of the dust; still, it works o.k.

If the Horizon works just as well dirty as clean, why not let it get dirty? First, it stands to reason that dirt acts as a heat insulator, and that dirty chips can't dissipate heat as efficiently as clean ones; therefore, clean chips should run cooler than dirty ones. Since heat is the enemy of the chip, clean chips should last longer than dirty ones. If the chips last twenty years rather than ten, so much the better.

Second, though I have never had any electrical problems caused by dirt in my Horizon, I have had problems with my terminal which went away when I cleaned away the dust. Presumably dust can conduct enough electricity to cause a partial short, thus in effect connecting together pins which should be kept isolated from one another.

In short, it seemed reasonable to do what I could to reduce the amount of dirt that got into my computer. Along came Steve Hogan with his Fan Filter Kit, which seemed just the thing. I was really worried about dirt; Steve was worried about heat; the fan filter kit was supposed to cure both problems.

The Kit came shortly after I ordered it, delivered by UPS. I ordered a new fan as well, since Steve concluded his article by recommending an upgrade to the Rotron Whisper XL model WX2H1 fan, "which delivers significantly more airflow than my old fan with only a slight increase in fan noise."

Re-Installing the Old Fan

The first thing I did was to try out the new fan, which was powerful, but so much noisier than my present fan, an older Rotron Whisper model, that I decided to keep my old fan in my Horizon. I'll use the XL instead to replace a fan on my hard disk system which is such a screamer

that I have had to put the box in a different room, connected to my computer with an 18' cable. Thus, as far as the current installation was concerned, I had only got the fan filter kit.

The Kit came with hardware, a filter box, a finger guard, several filters, and instructions. Being a true believer in the idea put forth by Robert Pirzig in Zen and the Art of Motorcycle Maintenance that a repairman should never read instructions but figure procedures out for himself, I tossed the instructions in the waste basket and proceeded with what seemed an obvious enough task.

The first chore was to disconnect the fan from its present location (I was smart enough to turn off the power first!) This was simple enough--just pull off the wires and unscrew the four bolts holding the fan in place. I did come to realize, however, that bottom right mounting bracket of the fan when it was inside the box (looking at the fan from the back of the computer) was not very accessible--I could just reach it from the mother-board side with the tip of one finger. What would I do if anything went wrong?

The four bolts were easy to unscrew, and the fan slipped right out. I cleaned it off with a paper towel (it was filthy!), then vacuumed out the area underneath the fan opening (and the rest of the computer as well, while I was at it). Reversing the fan was simple enough--an arrow indicates the direction of air-flow, practically eliminating the possibility of confusion.

Problems, problems

Now things got slightly sticky, however, in two respects:

1) The two wires leading to the fan weren't any too long in the first place; would they fit, with the fan in a different configuration? Fortunately, mine did--just. The fit was so snug that I had to use a needle-nose pliers to get the wires back on the exposed pins. (I realized that the fan is powered by A.C. current, so I didn't have to worry about which wire went to which pin.) I understand that some Horizons are wired differently, and that the old wires would have to be extended with a splice: not an impossible task, but one which I was happy to forget about.

2) Since there's virtually no access to the bottom right mounting bracket of the fan, I had trouble keeping the machine bolt positioned in the bracket so that it could be fed from the inside of the cabinet to the outside once the fan had been slipped into its place. If this concept is difficult to put in writing, it is even more difficult to accomplish in real life! In essence, I had to insert the bolt in the fan housing, then guide the fan carefully down into its "slot", match the bolt to the hole without letting the bolt touch the back plate, and then ease it through the hole without being able to use a finger to assist from the inside. Meanwhile (of course) the descent of the fan was being resisted by wires leading from the power supply. Fortunately only the one bolt required this gentle handling--the others could be slipped in afterwards, though the bottom left bolt was also slightly tricky.

Installing the filter

Once the bolts had been eased through the holes from the inside of the box, the rest seemed simple. I screwed on the four threaded spacers, slipped the filter over the spacers, positioned the aluminum filter frame over the filter, and screwed in the thumbscrews--but wait! The thumb-screws were too short. Steeeeeve, what have you done to me?

In desperation, I dug the directions out of the waste basket. Finally studying the assembly diagram, I discovered that I was supposed to have used the plastic finger-guard as a spacer between the back plate and the filter--this would make the spacers stand far enough away from the box to allow the thumbscrews to engage them properly. (Even though my Horizon already had a built-in finger guard rather than the simple large hole of some other models, I decided my fingers wouldn't object to this extra measure of protection.)

Putting in the plastic guard was extra trouble because the longer distance to the threaded spacers required the use of the next longer set of bolts supplied with the kit--so I had to remove the fan completely, and then insert it all over again. By this time, however, I had become experienced enough that it took only ten minutes. This time the thumb screws fit perfectly. Job well done.

Oh, no!

When I turned on the Horizon to enjoy watching the dirt begin to accumulate on the outside of the filter, the noise from the fan was deafening! What had gone wrong? A minute's inspection revealed that my old fan was not quite so easily reversed, since the blades now rubbed against the built-in finger-guard on the back panel. What to do? I realized that inserting washers between the fan and the panel would keep the blades from hitting against the panel, but would also allow some air to seep through from the box itself, thus reducing the efficiency of the fan. So I decided to make myself a gasket out of thin cardboard, using the fan itself as a guide. I did this, though of course I had to take the fan out once more--but I wasn't worried, because by now I had become an expert.

I came to realize an unanticipated advantage of using a gasket: by making the holes in the cardboard small enough so that the bolts had to be forced through the gasket stock, the gasket held the bolts in place while I slipped the fan back down into its slot. Three minutes after beginning the reassembly process (I'm going for the Guinness Book of Records!), I was in business. Fan worked. Filter worked. Dirt, look out!

The right way and the wrong way

Despite the trouble I got into from not reading the directions, I decided that Robert Pirzig was right. True, it did take me three tries, but if I had taken the time to read the instructions, I would have removed the entire back panel (held in place by **nine** machine screws), positioned the fan with the wires at the bottom right (where they would not be accessible from the top), retied the bundle of power supply wires, and replaced the back panel; and I would have had to go through this whole process twice, since the instructions said nothing about the possibility that the fan would rub against the finger guard. My way may not work for everyone, but it worked for me.

Moral: **Never read instructions.** (But keep them handy just in case this moral fails you.)

Maintenance

The one drawback I can think of to a filter is that it needs maintenance. After four years of totally disregarding maintenance, I now will have to check the filter at least once a month. Fortunately, my Horizon is situated so that I always see the filter before I see the front of the box. Still, I could imagine that for someone with a Horizon with its back to a wall, and who thinks, "Out of sight, out of mind," it would be better not to have a filter than to have one. Which reminds me that one of the few Horizons I have ever seen fail did so because it had been provided with filter which got totally clogged for lack of maintenance, and the whole system overheated.

Conclusion

Do I like my new filter? You bet! Does it do any good? I'll probably never know, but I'll sleep easier. Now, if you'll just excuse me, I'm going to go to the next room to install the new Rotron Fan and Sound Steward Filter on my hard disk drive ...

\$ \$ \$

Basic Kit: \$20.00; Hole-Plug Kit for the Horizon, \$7.00. Rotron Whisper XL Fan \$30.00.

The Sound Steward
1606 West Southgate Avenue
Fullerton, CA 92633
(714) 871-6636

#

WANTED!

I am very interested in acquiring a ram board for the Horizon. Write: Dr. Tom Trozera, 437 Pine Needles Dr., Del Mar, CA 92014

#

Lazy Man's Secret Handshake

By Warren Lambert

With a simple cable, assemblaphobes can help a buffered printer print full speed with any software by using secret handshakes already in the operating system.

Gary Sabot (1982) described how an ETX/ACK protocol enabled his buffered daisy wheel printer to print at maximum speed. Using this standard protocol, the computer transmits output in bursts terminated by a marker ("ETX"), and waits until the printer replies "ACK," which means: "I caught up; send me more." Without this "handshake," the printer cannot run full speed. A 55 character-per-second NEC SpinwriterTM, for example, can print at almost 600 baud, but only if its buffer is kept full so the poor machine never just sits there waiting for Baudot.

Transmissions at 300 baud (about 30 characters per second) are particularly limiting with fancy word processors, such as Micropro's WordStarTM which sends many control signals for microjustification, subscripting, and print enhancements in addition to printable text characters. Table 1 shows the relative time required to print three examples: a plain text, a graph with many spaces, and a printing test using all 127 characters on a technical-math thimble, using bold **face**, and other **SN^{AZZy}** enhancements.

Table 1

Relative time to print three pages:

Plain text, a graph with spaces,
and **ENHANCED** text
at 300 and 1200 baud

=====			
TYPE OF TEXT			
	plain	spacey	enhanced
RATE	TIME		
1200	100%	100%	100%
300	150%	318%	380%

The plain text was printed with a CP/M "ctrl-P TYPE" statement; the graph and complex text were printed with WordStar.

The loss of throughput is not so bad with plain text, but the loss was severe in the other two cases. At 1200 baud (about 120 characters per second) the Spinwriter's "logic seeking" zipped past blanks, so they took no time to print, and the print enhancement signals didn't waste time, since the buffer was full of printable data.

The ETX/ACK protocol is a good approach, but it involves patching the BIOS (basic input output subroutine) of the operating system to add the ETX/ACK handshake. Assembly language programming takes effort and rational thought, so I wanted to avoid patching my North Star Disk Operating System (N* DOS), CP/M, and Pascal, each of which (vexingly) has an original operating system and incompatible files. The following suggestion from MicroPro's WordStar Manual (1981, page D-6) is one of several options for installing WordStar for a NEC Spinwriter:

There is a hardware solution, requiring fabrication of a cable. It may not work in all cases; if it works in your case it has the advantage that it works with other software in addition to WordStar, and that no communications protocol, nor a printer driver that can input characters, need be specified in WordStar installation.

Make a cable adapter which runs the printer's 'Reverse Channel' signal (pin 19 of EIA connector) into the 'Request to send' line (pin 4 of the EIA connector) of the computer's serial interface. Check that the printer is set to produce a 'low' signal on buffer full...

Sadly, MicroPro's suggestion didn't work on my North Star, so I tried pin 20 (data terminal ready) instead of pin 4.

The rationale was monkeys, typewriters, and Hamlet, and it worked like a charm! The computer and printer handshake with CP/M, N* DOS, N* BASIC, WordStar, and N* Pascal. Using InfoSoft's CP/M version of N* Basic, I can even send console (#0) printout to the printer (#1) using the civilized software print switch (ctrl-P) of CP/M. With any of the software mentioned, the Spinwriter can interrupt printing instantly and resume later with no loss of characters.

Why does it work?

The BIOS in Lifeboat CP/M for N* contains a printout subroutine with the mnemonically memorable name "HOROUT1."

```

HOROUT 1:
  IN      5          input port
                    status1
  ANI     1          get the status
                    bit
  JZ      HOROUT1   loop back if
                    unready
  MVI     A,C        otherwise, move
                    to A
  OUT     4          send byte out
                    port 4
  RET                                go back where
                    you came

```

This routine says, "Check the printer status port, and output only if the printer is ready. If the printer is busy, just sitspin in a 37 T-cycle loop 108,108.108 times per second (about 65 million r.p.m.) until the printer is ready." The Spinwriter's pin 19 produces a "terminal ready" signal which the standard BIOS handshake was ready to use without modification.

If you think this theory works too easily for the suffering involved, you're not alone. I called the North Star hotline to check it out with somebody who knew what they were doing. A nice lady said she was very sorry but they don't talk to computer owners, and if I wasn't a dealer or OEM, her lips were sealed. Then she hung up.

Inspired with self-reliance, I tested the theory by deleting "JZ HOROUT1" from the CP/M installation file, "USER.ASM" and assembled a new version of the CP/M operating system just like the original, except for that one missing line

'jump to horout1.' I used CP/M to "TYPE" a text simultaneously on the console and printer at 120 characters per second. There were no printer pauses in the output, and the Spinwriter was quickly overrun, printing confused garbage rather than the neat, well organized garbage I wanted. But when I switched both the console and printer ports to 30 characters per second, everything worked fine. No handshaking was needed. Jump-on-zero to HOROUT1 is the secret handshake!

Once a soldered cable showed that connecting NEC pin 19 to N* pin 20 Spinwriter worked, pin 19 was relocated to position 20 on the movable 25 pin plug that comes with the new 77xx Spinwriters; no soldering or special cables were needed. Just plug the printer into the computer and everything works fine, regardless of whether the user knows the secret handshake of the standard BIOS.

Reverting to 300 baud

I often use my HorizonTM to print material transmitted over the phone by the University mainframe computers at 300 baud using the Horizon-as-terminal. To do this, I switch the computer and printer to 30 characters per second. The Spinwriter came with switches for this purpose, but the N* needed a little rewiring using the "baud rate switch" suggested in the May 1978 issue of the late North Star Newsletter. (See also "Easy-Switch" articles in Compass, Vol. I, no. 4, and Vol. II, no. 1--Ed.) This switch lets you select two baud rates for a serial port. To make this switch, you connect two baud rates (e.g. 1200 and 300) according to Northstar's instructions (Northstar, 1978, p. 75) interposing a single pole single throw switch to keep one or the other rate clocks disconnected at all times.

There is nothing about the hardware approach to using the secret handshakes of your BIOS that is unique to people (like Atlas) with Horizons on their NECs; the concept should work with many computers with selectable baud rates, a "terminal ready?" test in the BIOS, and implementation of an "terminal ready?" port. The printer should have a buffer with an "I'm getting full" signal prior to data loss from overflow.

I prefer MicroPro's cable approach to the ETX/ACK protocol because I can use

software and operating systems right out of the box. It's easy, it takes no brains or effort, and it works.

REFERENCES

Lifeboat Associates. **MANUAL CP/M2 ON NORTH STAR DISK DOUBLE DENSITY-QUAD CAPACITY.** User's Notes, revision 1.1. New York: Lifeboat, 1979.

MicroPro International Corporation. **WORDSTAR INSTALLATION MANUAL, WS-3630-1(3.0).** San Rafael: MicroPro, 1981.

North Star Computers Inc. **HORIZON COMPUTER SYSTEM DOUBLE DENSITY, HRZ-D-DOC,** revision 1. Berkeley: North Star Computers, 1978.

Sabot, Gary. Speed up your Diablo printer. **MICROSYSTEMS,** 1982, 3(2), 72-75.

ENDNOTES

¹According to North Star (1978, page 72) the 16 pin header at location 3D on the backplane must be fully configured, not abbreviated, for the control signal ports to work.

²'WordStar' is a trademark of MicroPro International; 'Spinwriter,' of NEC Information Systems; and 'Horizon,' of North Star Computers.

#

N*SORT with BASIC 5.5

By Jim Lind

N*SORT and N*BUS are machine code patches to North Star Basic designed by SZ Software Systems. N*SORT allows the programmer to sort one- and two-dimensional arrays or string data with a single BASIC statement and N*BUS is a global editor. Each package has a program to personalize the host BASIC. The personalization depends on whether key addresses of subroutines within BASIC are found by the programs provided.

When INSUA made the latest version of NorthStar DOS and BASIC available, I tried the personalization program, SORTGEN, to incorporate N*SORT but it failed. EDGEN, the program provided to personalize N*BUS, worked fine.

Table A-1 of the N*SORT manual (Release 1.0 dated January 15,1981) provides the BASIC Templates of release 5.0. When these Templates were compared with the code of BASIC 5.5, the second and sixth were different. The following Templates apply to the latest BASIC:

BASIC 5.5 JMP ADDRESS FOR "SYNTAX ERROR" & LET ENTRY

1C4B	D2 04 13	JNC	1304H	<==SYNTAX O2=1
1C4E	C3 54 1C	JMP	1C54H	
1C51	11 50 24	LXI	D,2450H	<==LET ENTRY O2=7
1C54	CD 1D 2E	CALL	2E1DH	
1C57	E6 7F	ANI	7FH	
1C59	07	RLC		
1C5A	6F	MOV	L,A	
1C5B	26 00	MVI	H,00H	
1C5D	19	DAD	D	
1C5E	CD 9A 2E	CALL	2E9AH	
1C61	E9	PCHL		

BASIC 5.5 SUBROUTINES GETPTR & PGMPTR (Fetch program pointer)

```

2E2B 2A 2F 41      LHL  412FH      <==GETPTR O2=0
2E2E 3E 20        MVI  A,20H
2E30 2B           DCX  H
2E31 23           INX  H
2E32 BE          CMP  M
2E33 CA 31 2E     JZ   2E31H
2E36 7E          MOV  A,M
2E37 22 2F 41     SHLD 412FH
2E3A C9          RET
2E3B 2A 2F 41     LHL  412FH      <==PGMPTR
2E3E 7E          MOV  A,M
2E3F 23           INX  H
2E40 FE 20       CPI  20H
2E42 CA 3E 2E     JZ   2E3EH
2E45 22 2F 41     SHLD 412FH
2E48 C9          RET

```

A copy was made of the N*SORT disk. It is never a good idea to use the Master Disk in case something goes wrong. Then four lines of the BASIC program SORTGEN were changed to read as follows:

```
1550 DATA 210,-1,-1,195,-1,-1,17,-1,-1,205,-1,-1,230,127,7,111,38,0,25,205
```

```
1590 T1=B1+7\T2=EXAM(T1)\T3=EXAM(T1+1)
```

```
1740 DATA 42,-1,-1,62,32,43,35,190,202,-1,-1,126,34,-1,-1,201,42,-1,-1,126
```

```
1780 A$="PGMPTR"\O1=8\B1=B1+16\GOSUB2990
```

The "QUICK PERSONALIZATION" was used to build a TANDEM version for BASIC Version 5.5 at 1000H. The following addresses are provided for reference:

```

BOUNDS ADDRESS = 12BFH
SYNTAX ADDRESS = 1304H
COMPARE ADDRESS = 26D3H
VARPTR ADDRESS = 2B68H
GETPTR ADDRESS = 2E2BH
PGMPTR ADDRESS = 2E3EH
GETVAL ADDRESS = 307AH

```

N*SORT starting address: 475AH

```

FILL 4102,153
FILL 4103,74
FILL 9296,90
FILL 9297,71
FILL 9361,83
FILL 9362,82

```

The program TEST1 provided on the N*SORT disk should be run to make sure there are no errors. Good Luck!

#

Commentary

By Warren Saunders

I would like to submit the following comments on Compass, Vol. 4, no. 1):

1. "The Phantom Signal" - About 3 years ago I discussed this scheme with a North Star Engineer. While it does work most of the time under most circumstances, it is not fool-proof and therefore not officially endorsed by North Star. The reason is that the source signal (Output LATCH ENABLE) being applied to the phantom line is asserted very late in the memory-read cycle. It's the signal that enables the tri-state bus driver. In other words, the RAM board at the corresponding address location goes through a full memory fetch cycle (better not have wait states enabled on this board) up to the time that it would tri-state its data on the data bus. Then along comes the phantom signal from the controller board to abort the read process. The phantom signal just may not make it in time. Comments would be welcome from design engineers who know more than I as to how reliable this fix really is.

2. "Computer Pollution Solution" - I have just recently completed a similar fan reversal & filter installation modification to two Horizons. I was not ambitious enough to construct my own filter cover and select an appropriate filter material so I kept looking for a commercially available product and source of supply. After years (really) of looking I located a source. It's available from ACTIVE, PO Box 8000, Westborough, MA, 01581 (800) 343:0874.

They also have stores in Boston and Seattle. Part Number is 3-90-8099, cost is \$4.40 each. A replacement filter (3-90-8106) is also available for \$2.45. These filters are made exactly for these standard 4-11/16" fans.

ACTIVE'S catalog shows an exceptionally complete product line for computer and electronic parts. I've had no problems with several mail orders.

Only word of caution about the installation process: the older Horizons had a single large round fan hole (about 4") with a separate finger guard grill mounted on the outside. These are easy to convert, just turn the fan around and replace the grill with the filter assembly; you may need slightly longer screws though. However, if you have one of the (newer?) Horizons with lots of small stamped-out holes in the back panel for a grille, you may be in for more work. I seems that on some of the fans the center on the air intake side sticks out a bit further than the sides. Thus when turned around the center of the fan will press against the back panel and will not rotate! You'll need to either insert standoffs or cut away the stamped-out holes making a single large hole. I elected the latter since opening up the stamped-out holes makes room for a lot more air flow. Thus the air flow loss through the filter will not be significant. By the way, if you're the lazy type, plain old vinyl tape will plug up all the back panel holes just as effectively as plastic plugs and specifically-cut sheet metal.

#

Here's the Meat!

In last week's issue, we printed "Accessing the HDOS File Manager from North Star BASIC, by Terry Hough. But in the haste to paste up the issue, we forgot the listings! With our apologies to Terry, we now present the meat:

```

13000 REM *** FMSDV      *** 101483 *** EXECUTE FILE MANAGER FMSDV
13010 REM *** LOADER    *** 101483 TW HOUGH
13020 REM ----- THEORY OF OPERATION -----
13030 REM LOAD A SHORT MACHINE LANGUAGE PROGRAM AT 64H OR 100D
13040 REM BY FILLING MEMORY BYTE BY BYTE AND "CALL" THE PROGRAM.
13050 REM THE ROUTINE PRESERVES CRITICAL REGISTERS AND EXECUTES A
13060 REM SAFETY RETURN (C9) TO REENTER THE CALLING BASIC PROGRAM.
13070 REM -----
13080 REM \                SET DEFAULT ACCOUNT NUMBER                \
13090 REM \ DECIMAL      \      \ MACHINE      \                ACTION      \
13100 DATA 06,101      \REM - 3E 65      - LOAD A WITH DRIVE NUMBER
13110 DATA 33,03,00    \REM - 21 03 00    - LOAD HL = ACCOUNT ID # (1-255)
13120 DATA 62,17      \REM - 3E 11      - LOAD A WITH FMSDV POINTER
13130 DATA 195,74,01  \REM - C3 4A 01    - JUMP TO FILE MANAGER
13140 DATA 00          \REM - 00          - NO OPERATION
13150 DATA 201        \REM - C9          - SAFETY RETURN TO CALLING PROGRAM
13160 DATA 00          \REM - 00          - NO OPERATION
13170 REM -----
13180 Q=100             \REM                - SET ORIGIN TO 64H OR 100D
13190 FOR I=0 TO 12\READ X\FILL O+I,X\NEXT
13200 FILL (O+3),A      \REM                - SET DEFAULT ACCOUNT ID #
13210 Q=CALL(O)         \REM                - JUMP TO MACHINE LANGUAGE PROGRAM
13220 RETURN            \REM                - FOR SUBROUTINE USE
13230 REM -----
13000 REM *** FMLX      *** 111183 *** EXECUTE FILE MANAGER FMLX
13010 REM *** LOADER    *** 111183 TW HOUGH
13030 REM ----- THEORY OF OPERATION -----
13050 REM LOAD A SHORT MACHINE LANGUAGE PROGRAM AT 64H OR 100D
13060 REM BY FILLING MEMORY BYTE BY BYTE AND "CALL" THE PROGRAM.
13070 REM THE ROUTINE PRESERVES CRITICAL REGISTERS AND EXECUTES A
13080 REM SAFETY RETURN (C9) TO REENTER THE CALLING BASIC PROGRAM.
13090 REM -----
13100 REM \                LOAD & EXECUTE A TYPE 1 FILE THRU HDOS 2.2.0H      \
13110 REM \ DECIMAL      \      \ MACHINE      \                ACTION      \
13120 DATA 33,111,0    \REM - 21 6F 00    - LOAD HL = PATHNAME ADDRESS 6FH
13130 DATA 6,0         \REM - 06 00      - LOAD B WITH FMLX OPTION CODE
13140 DATA 62,13      \REM - 3E 0D      - LOAD A WITH FMLX POINTER
13150 DATA 195,74,1   \REM - C3 4A 01    - JUMP TO FILE MANAGER
13160 DATA 201        \REM - C9          - SAFETY RETURN TO CALLING PROGRAM
13170 REM -----
13180 Q=100             \REM                - SET ORIGIN TO 64H OR 100D
13190 DIM F$(29),F1$(29)\F$="TSS/C,SYSTEM,101"+F1$
13200 REM - OPTIONAL - INPUT "ENTER TYPE 1 PATHNAME : ",F$;F$=F$+F1$
13210 FOR I=0 TO 10\READ X\FILL O+I,X\NEXT\O=O+I-1
13220 FOR I=1 TO LEN(F$)\FILL O+I,ASC(F$(I,I))\NEXT
13230 Q=CALL(100)       \REM                - JUMP TO MACHINE LANGUAGE PROGRAM
13240 RETURN            \REM                - FOR SUBROUTINE USE
13250 REM -----

```

Soft-Sector on the Horizon

By Ted Carnevale
Neurology Dept., SUNY
HSC T12 Rm020
Stony Brook, N.Y. 11794

I share O. C. Stokes's interest in the use of other disk formats and CPU's with the Horizon (see Compass, Vol. IV, no. 4, p 33). The Z80 hasn't faded into oblivion yet, but I suspect that the days of hard-sectored disk formats are numbered.

So I decided to add the ability to handle soft-sectored formats. The next question was: 8" or 5.25"? Any change would mean purchase of new equipment, and probably the investment of a fair amount of time cobbling up a BIOS that could deal with both hard- and soft-sectored formats (I have several hundred hard-sectored disks full of data, and I don't want to just throw them all out or pay to have them copied).

Adding soft-sectored 8" drives meant buying a new controller card plus a self contained box with power supply for the drives. This would cost over \$1000, if quality dual-sided drives were used. The advantages to this approach are disk capacity and I/O speed (big DSDD floppies can hold a megabyte or more, and can transfer data twice as fast as 5.25"), and format compatibility with other CP/M machines using 8" drives (through the IBM SSSD format).

There are two ways to add 5.25" soft-sectored format capability: by purchasing a second disk controller card and permanently attaching it to one or more drives dedicated to soft-sectored formats, OR by using a controller card capable of handling BOTH soft- and hard-sectored formats. As far as I know, there is only one disk controller that can do the latter: the Morrow DJ/DMA, which can run four 5.25" and four 8" drives simultaneously. The BIOS provided with this card can read and write N* 35 track hard-sectored disks. This is the way I decided to go.

It has not been smooth sailing with this board, however. Several hardware and software problems had to be dealt with. First of all, the N* bus grounds one of the lines needed for 24-bit addressing (line 61, used for A20), so the DJ/DMA

card must run in its 16-bit address mode. Second, North Star's dynamic RAM boards cannot be used with the DJ/DMA--the disk controller's direct memory access operation with the RAM refresh cycles--so static RAM must be used. A 64K static RAM board costs about \$300-400 (I use the Fulcrum board).

Installing CP/M 2.2 to use this controller has been an educational experience. The BIOS supplied with the DJ/DMA must be assembled with your own copy of Digital Research's MAC or RMAC--Morrow doesn't include either of these with their BIOS. Conditional assembly of large sections of the BIOS is controlled by tables of options that specify the hardware configuration of the system for which it is intended.

The BIOS looks like it should run with little or no change on a "standard" Horizon, but it needed extensive revision for my particular collection of I/O devices. The drive parameter blocks and format bytes for 35- and 40-track hard-sectored formats also needed revision to work properly.

For a while, I kept both the N* and the Morrow controllers in my machine, with the Morrow connected to an outboard disk drive. This way I could boot up a 64K CP/M with the N* controller and use it for development and testing of the Morrow BIOS. I created and revised multiple buggy revisions of a 40K Morrow CP/M that I loaded and ran under DDT until one of them finally worked.

At last I was able to remove the N* controller board and let the DJ/DMA handle cold boots. This meant that the NorthStar CPU card's "auto-jump" feature that starts the N* disk controller on cold boot had to be disabled by removing the 74LS175 IC from position 4G. Aside from this, no other changes were needed to make this venerable Z80 board work with the DJ/DMA.

At this point I have been using the DJ/DMA for about three weeks, and I'm

generally pleased with its performance. It does indeed read and write N* format hard-sectored disks, as well as Morrow-format soft-sectored floppies (more about this below). I have revised the BIOS again, adding full track-buffered disk i/o, which takes up an extra 5K of main memory. This speeds up most disk accesses considerably. I'm in the process of modifying the BIOS so that it will use two track buffers, one for disk reads and the other for disk writes, which will improve speed further. Both of these buffers will be located in auxiliary (banked) memory, and only a 128 byte buffer for logical sectors will remain in main memory.

This has not been as easy as it sounds (does it sound easy?). Morrow's BIOS source code is enormous (over 180K). Much of this is conditionally-assembled stuff that doesn't pertain unless you use other Morrow hardware (I/O cards or their other disk controllers). By careful editing, I was able to reduce the size of the BIOS source to about 88K. This took some doing, but the result was a file that was much less cumbersome.

There are a few problems with the DJ/DMA. First, the format program supplied by Morrow (FDJ.COM) doesn't work with soft-sectored disks. Its main menu provides soft-sectored formats as an option, but the program returns with an error message when this choice is selected. However, the controller seems to be able to read and write soft-sectored Morrow-format 5.25" disks. I have been reduced to formatting soft-sectored disks on a Kaypro using a multiple disk format program!

The second problem is that SYSGEN.COM doesn't always work right. Sometimes it damages the first sector on track 0, overwriting the byte that tells the BIOS the format of the disk. When this happens, the disk cannot be logged in and any attempt to access the data on it with the usual CP/M functions and utilities will produce an error message--you can't even get a directory of the disk! This happens even with disks that were formatted by the DJ/DMA using FDJ.COM.

A disk that has been damaged in this way can be recovered by using the public-domain disk utility DU-Vxx.COM to read track 0 sector 1 of an undamaged disk, then swapping disks and writing this sector

to the disk that SYSGEN had munged. After exiting DU-Vxx, SYSGEN can be run again. This time it may show an error message or two when writing to the systems tracks that you just fixed. If this happens, telling it to ignore the error and then making it write to the same disk again will usually work.

Finally, the DJ/DMA's documentation states that this board (which has its own Z80 and on-board RAM) can be programmed to create and read/write other soft-sectored formats. However, no further information is provided about how to do this, and the manual that is supposed to explain these details (which their documentation said would be released in 1982) has never been published. I have called Morrow Designs several times trying to obtain additional information on this point, with no success--except to find out that the DJ/DMA is used successfully for just this purpose in their new Tricep computer (a multi-user machine that runs a version of UNIX). Just the other day, I saw a notice in the back pages of BYTE from another DJ/DMA user who wants to find out how to handle non-Morrow soft-sectored formats.

Well, does any other N* user have answers to these problems? If anyone else is using the DJ/DMA, I'd be glad to exchange notes with them.

#

FOR SALE!

COMPUTER: 64K North Star Horizon w/ 2 quad disk drives, TeleVideo 950 terminal, built-in PMMI modem plus much software. Price: \$1500. (Price includes CP/M software worth over \$2000.) Also optionally available: NEC 3515 Spin Writer letter-quality printer w/ tractor. Price: \$1200. All equipment, etc. excellent condition. (202) 362-9854.

#

Horizon Database Management Program. DD/SD. \$13.00 File size limited by available disk space. Uses hashing for fast search of file. Options: Create, Add, Look-up, Delete, Change, Reports. Roger Pollock, 2315 Lee Ave, Melbourne, FL, 32901

Computer Books

By Charles A. Hightower
P.O. Box 3276
Montgomery, AL 36109

Since computers have become so popular lately, the prices of computer books have gone up a great deal. Today you can end up paying fifteen or twenty dollars for a **lousy** computer book. The good ones cost the same, but are harder to find.

I've bought four books on Z80 Assembly Language over the past couple of years, and will review them on the following criteria on a scale of one to ten.

FAMILIARITY LEVEL

Novice (1) to Expert (10)

Some books take you through the basics - how to add in Binary, Hex, and so on. Other books assume that you are already familiar with Assembly Language and that you just want to learn the Z80 brand.

READABILITY

Confusing (1) to Well Written (10)

How well does the book flow from chapter to chapter? Are the chapters written for humans or for some higher life form?

MEANINGFUL EXAMPLES / COMMAND EXPLANATIONS

Lost me somewhere (1) to Easily understood (10)

When a command or series of commands are executed, what action takes place? What are the resultant register values? Most of the books I've come across do not explain the commands in words. Instead, they provide an illustration with boxes and arrows pointing in all directions, with notes on how many clock cycles the command requires. Since I am a programmer and not an electronic engineer or chip designer, these diagrams don't do me any good. When I flip through a book, if I see very many of the arrow type examples the book goes back on the shelf.

APPENDICES

What appendices? (1) to Well planned (10)

Did the author overlook any useful appendices? Did the book have any unexpected appendices? I expect an Assembly Language book to have a reference chart for assembly commands listed in alphabetic and numeric order. I also expect to see an ASCII chart so that I don't have to dig through my North Star manual to find a character's value in decimal or hex.

LOCATING COMMANDS

Hard to find (1) to It's a snap (10)

Can an unfamiliar command be found any way other than reading the book cover to cover? What about reading up on a task - such as a block move, can the associated commands be easily located?

% % %

Book: Machine and Assembly Language Programming 199 pages

Author: David C. Alexander

Publisher: TAB Books, Inc.

Familiarity Level: 3

Readability: 8

Meaningful Examples / Command Explanations: 7

Appendices: 8

Locating Commands: 8

The book is easy to read and has many good examples. The book falls short, however on its explanation of what the commands actually cause to happen. Many times I was left wondering what registers or memory locations were affected. On appendices, the book meets my expectations and has a handy appendix on displacement jump values.

All in all, the book is helpful. I suppose the author had to add "Machine Language" to the title to make it original,

because little mention was given to machine language other than the fact that computers work in it. I recommend this book for anyone familiar with computers who would like to get started in Assembly Language.

% % %

Book: 8080/Z80 Assembly Language 318 pages
Author: Alan R. Miller
Publisher: John Wiley & Sons
Familiarity Level: 5
Readability: 9
Meaningful Examples / Command Explanations: 3
Appendices: 9
Locating Commands: 4

As the title implies, the book is written for both 8080 and Z80, with about 75% and 25% respectively of the book dealing with each. As for the appendices, the author went overboard. On appendices, all my expectations were met, plus an 8080/Z80 cross-reference chart. The author even included a 64K memory map.

The book starts out assuming the reader knows nothing about computers or number systems. The first couple of chapters leads the reader by the hand through good explanations of number systems. Right after that, the reader is left on his own in the middle of assembly language examples. To be honest, this was my first assembly language book and I had a hard time following that transition. I read over those first few chapters many times before the (logic?) of the change in pace sunk in. After the transition the examples are sketchy and difficult to follow. I recommend this book for someone who wants an 8080 reference manual as it has little use as a Z80 manual.

% % %

Book: Z80 Assembly Language Subroutines 497 pages
Author: Lance A. Leventhal & Winthrop Saville
Publisher: Osborne/McGraw-Hill
Familiarity Level: 6
Readability: 9

Meaningful Examples / Command Explanations: 8
Appendices: 3
Locating Commands: 6

The book is well-documented example after example. On appendices, the author included a partial ASCII chart, but little else. I recommend this book as a reference manual for documented common routines, and command descriptions (if you can find them).

% % %

Book: Z80 Users Manual 326 pages
Author: Joseph J. Carr
Publisher: Reston Publishing Company, Inc.
Familiarity Level: 9
Readability: 5
Meaningful Examples / Command Explanations: 8
Appendices: 1
Locating Commands: 9

This book gets very deep, very fast. True, the author does include some of the "box and arrow" diagrams. The reason I bought the book, though, was because of its complete explanations of commands. The last 180 pages are an alphabetical listing of each and every command with a brief discussion on what registers, flags, and memory locations that are affected by the command. I don't mean just every reserved word; I mean every permutation of every command. On appendices, the book has none. I would have given them a zero, but a one was the lowest I was allowed. I recommend this book as an advanced programmer's reference manual. Even then, a straight read-thru is difficult.

#

Vendors Column

In this column we reprint vendors' descriptions of their own products. Of course, INSUA makes no claims about the quality or merchantability of any commercial software, hardware, or services. **Caveat emptor!**

INSUA:

We had planned a letter to S-100 oriented suppliers, but this is not the letter we intended to write. The letter we had planned would have stated that we are another technical magazine for the advanced computerist, but after the recent demise of Microsystems we may be the only technical magazine for this audience.

We cover a wide variety of topics, and while we are not machine or system specific, the majority of our readers are using S-100 systems because it is so well suited for our applications. The rest of our readers are split between the Apple II and the IBM-PC with a few using TS100, VIC-20, or C-64 units especially for dedicated microcontrollers. Many of our readers are working with several different systems.

The Computer Journal is published for the advanced, technically oriented people who are involved in building, interfacing, and designing microcomputer systems. Our readers are tired of product review articles which are full of advertising hype, and read TCJ for the advanced technical information which we offer.

We don't cover games, checkbook balancing programs, or business programs, and rarely publish product reviews unless they are of specific interest to our audience; and then we usually publish an example of an application instead of the usual type of review. What we do publish is information on interfacing, measurement, control, robotics, operating systems enhancement, programming, and construction projects.

R. Arthur Carlson
Publisher
The Computer Journal
11003 South Main, PO Box 1697
Kalispell, MT 59903-1697

\$ \$ \$

INSUA:

I would like to offer the readers of The Compass the opportunity to produce professional looking manuals, brochures, books, etc. by having them typeset at a cost about 40 percent below the lowest typesetting charges. This can be done because I use both Advantages and Horizons in my typesetting business. Usually about 60 percent of typesetting charges are allocated to re-inputting the manuscript material into the computer system. As I use Northstar hardware, I would be able to use the material submitted on the disk and thus eliminate the input costs.

If any subscriber has text that is intended to be reproduced, I would like to hear from him or her with a sample disk, printout, and specifications of the typeset material.

Write: Lee D. Quinn
20 Lincroft Avenue
Old Bridge, NJ, 08857
(201) 679-6279.

\$ \$ \$

Letters to Joe

By Joe Maguire

Several readers have written me for help with various problems. What follows is a sampling of those of general interest:

Dear Joe: I copied this program out of a magazine but I can't get it to run on my Horizon. (listing inclosed) E.Y.

(The listing enclosed by E.Y. was of a small utility program written in Z80 code.)

Dear E.Y.: The problem with your program is that you are trying to load and run it at the wrong address. It is true that some programs written in Z80 code are "address independent" (meaning that they can be loaded and run at any address) but your example is not. To be address independent the program must be coded using the Z80's relative jump instructions. No "calls" can be used since they always reference a specific address. Look at the "ORG" in the listing. It shows that the program origin is FE00. Load it at that address and it should run OK.

Dear Joe: I want to edit the file "EQUUS" that came on INSUA disk #1024. I have CP/M and WordStar. How can I get the file so WS can edit it? PUZZLED in LA.

Dear PUZZLED: In order to get EQUUS onto a CP/M disk where WS can get at it you will have to use DOS and DDT. Do the following:

* Using DOS, load the file EQUUS at some address above DOS. LF EQUUS 2000

* Now calculate where the file will end in memory by multiplying the number of disk blocks shown in the directory by 256 and adding that to 2000. This is a hex calculation.

$$52 \times 256 = 13312 = 3400H + 2000H = 5400H$$

* Bootup CP/M and run DDT. (Don't turn the computer off for this step, use the DOS command: JP E800) DDT will have wiped out the memory between 100H and 1400H while it was loading and moving itself out of the way so that was another reason for

selecting the original load address of 2000H.

* Using the memory move command of DDT, move the file EQUUS from its resting place at 2000H to 100H where it can be saved by the CP/M SAVE command.

M2000,5400,100

(DDT has moved itself to high memory and the area from 100H up is available now.)

* Do a Ctrl/C to go from DDT back to the CP/M prompt and use the SAVE command.

SAVE 52 EQUUS.SYM

Notice that the number of blocks to save under CP/M is the same number of blocks that the file occupied in the DOS directory. The suffix SYM is added to remind you that this file is actually a symbol table. It is entirely optional.

Bring up WordStar and request to edit the file EQUUS.SYM in your usual fashion. You are on your way!

Dear Joe: I have a program on an old single density disk which I want to run with DOS 2.1.1. I load it at 2A00 but so far I can't get it to work properly. NO LUCK in Saskatchewan.

Dear NO LUCK: When trying to convert programs from one version of DOS to another, try this:

* Bootup DOS 2.1.1 at 100H.
* GO the Monitor program at 1000H

GO M1000

* Using the MM (move memory) command of the Monitor, move a copy of the front end of DOS to 2000H where your single density program expects it.

MM 100,5F 2000

Remember that all I/O and disk

operations go through the jump vectors located in the first 60 bytes or so of the DOS. We moved an extra 30 just to make sure we got 'em all.

* Jump to the start of your program and see what happens.

JP 2A00

If you get the signon message, or whatever you are supposed to get, you have isolated the problem to the I/O jumps in the old program. Just change them all from "20" to "01".

But why does this work now? Well, examine the jump vectors you moved to 2000H. Let's look at COUT (character output) as an example.

The original jump to COUT is located at 010DH. A copy of the jump to COUT is located at 200D. They both point to the same address in the DOS I/O block.

```
010D C3 DF 07
200D C3 DF 07
```

(In DOS 2.1.1, COUT goes through an internal DOS routine first but eventually gets to the I/O block.)

When your old program jumps to the COUT routine, it expects to find it at the old DOS location of 200D. By placing a copy of the correct jump to COUT for DOS 2.1.1 there, you have re-directed the output to the proper place. It will work the same for all of the other jumps to CIN, DCOM, etc. since in all versions of the DOS they are in the same relative location.

One problem with old single density programs: If they do any disk primitive operations, such as file creation, you are going to have problems. When double density was introduced, the arguments to be passed in the BC register were upgraded to include a density flag. Old programs must be modified to have the correct arguments or file operations will not work properly.

Dear Joe: I liked your "Super Basic" patch for Basic 5.2 (Compass Vol. 1, No. 2, p.14) and would like to use it with Basic 5.5.0. Have you modified Basic 5.5.0 yet? T.S., Wisconsin

Dear T.S.: Unfortunately, I haven't but I will tell you how you can do it. The process is basically the same for modifying any program.

You will need a disassembler program. There are two in the INSUA disk library. RESOURCE on disk #1014 works with CP/M and, in my opinion, is excellent. Disk #1006 has one written by Lance Rose which will work with DOS but I haven't tried it. To get Basic 5.5.0 onto a CP/M disk, follow the procedure given for transferring EQUUS above.

With either disassembler, do the following:

* With the patch installed in the original program, disassemble the areas around the patch locations. There is no need to disassemble the whole program.

* Note the code sequences around the patches.

* Disassemble portions of the new program around approximately the same address locations. Adjust the addresses for different program origins if necessary.

* Try to spot the same code sequences in the new program and, if you can, insert the patches in the same relative locations.

Most updates of any program change only small portions of the code. This may shift the addresses of other routines up or down somewhat but the code patterns will be the same. The absolute address references will be different, of course, but if you see, for example, three CALLs in a row followed by two JZs and a RET, you should be able to spot the same pattern in the other program without difficulty. This is how I did it when I transferred my Two Column List routine (Compass Vol. 3, No. 4, p.25) to DOS 2.1.1. Good Luck!

Note to Readers: I will be glad to try and answer similar questions to those above. I do not, however, have any of North Stars programs other than DOS or Basic. Questions about specific items of North Star software should perhaps be directed instead to INSUA.

Write: Joe Maguire
2321 Foxhall Dr.
Anchorage, AK 99504

Letters to the Editor

INSUA:

I have read volume 4 number 3 of The Compass and wish to make several comments.

First, I would like to offer my services to answer readers' questions regarding using computers as input to typesetting systems. I do not have any great competence in the internal architecture of computers but I do have special insight into applications as they relate to typography.

Next, for the third time, I would like to suggest that your magazine would be easier to read, be professional looking and more economical if it were set in type. The first two attributes are fairly obvious. Economy is afforded because you can have about 30 percent more text in the same number of pages. This means that your printing costs (and, perhaps, mailing costs) would be reduced.

In closing, I would like to be afforded the same opportunity as Randy Fisher, in publishing the material which appears below in the "Vendors Column." (Which should really read: "Vendor's Column.")

Lee D. Quinn
20 Lincroft Ave.
Old Bridge, NJ 08857

(INSUA has considered having Compass type-set, but so far has resisted the temptation. For the editor, the principal advantage of using our own printer is a turn-around time of about two minutes per article. The best turn-around time we could hope for from even a local printer would be two days; from a distant printer it might be two weeks. We do not mean this in any way as a criticism of the excellent services offered by many printers who will now accept material on disk or via modem. If anything, it is a reflection of our inability to do anything except under the compulsion of an impending deadline.

Moreover, by using a proportional-spacing program like MagicBind, we do achieve a great deal of condensation of text; to achieve much more, we would

have to use smaller type, or do photo-reductions.

Yes, you do have a full right to appear in the "Vendors Column"--and you're there! Our nit-picking reply to your correction of our punctuation is that the proper punctuation should be "Vendors' Column," that is, plural rather than singular, except that we can never know in advance whether there will be one vendor per issue or more than one. For the same reason, INSUA is the International North Star Users Association--since we can't decide whether we should write "User's" or "Users'," we just leave the apostrophe out altogether. --Ed.)

INSUA:

Thank you for the Modem 712 program. I have only noticed two shortcomings of the distribution disk. The documentation for the program was not included, but it is available from Elliam Associates on SIG/M User's Group volume 139 or volume 168. Also, volume 119 has a program CHGLIB14 that makes it very easy to change the phone number list. The second problem is that the printer toggle does not work on the Advantage. The program apparently thinks the printer is in slot one, as normally configured. But if you have a parallel printer, slot two is also a possible arrangement for the printer. I don't believe the overlay for the Advantage has a place to indicate the printer port.

Another observation is that with the Smartmodem, a hangup while dialing can be done by simply pressing the space bar in Terminal mode. This is much faster than control-N.

Thanks again for the program.

Sincerely,
Kenneth O. Henderson

FCS

Fischer Computer Systems

445 Bay Street, Angwin, CA 94508 (707) 965-2414

Specializing in North Star Computers
Horizon and Advantage
Service and Upgrading
NX Dos TurboDos CP/M
operating systems supported

Special SALE

Reconditioned North Star Horizons and Advantages any configuration.

Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

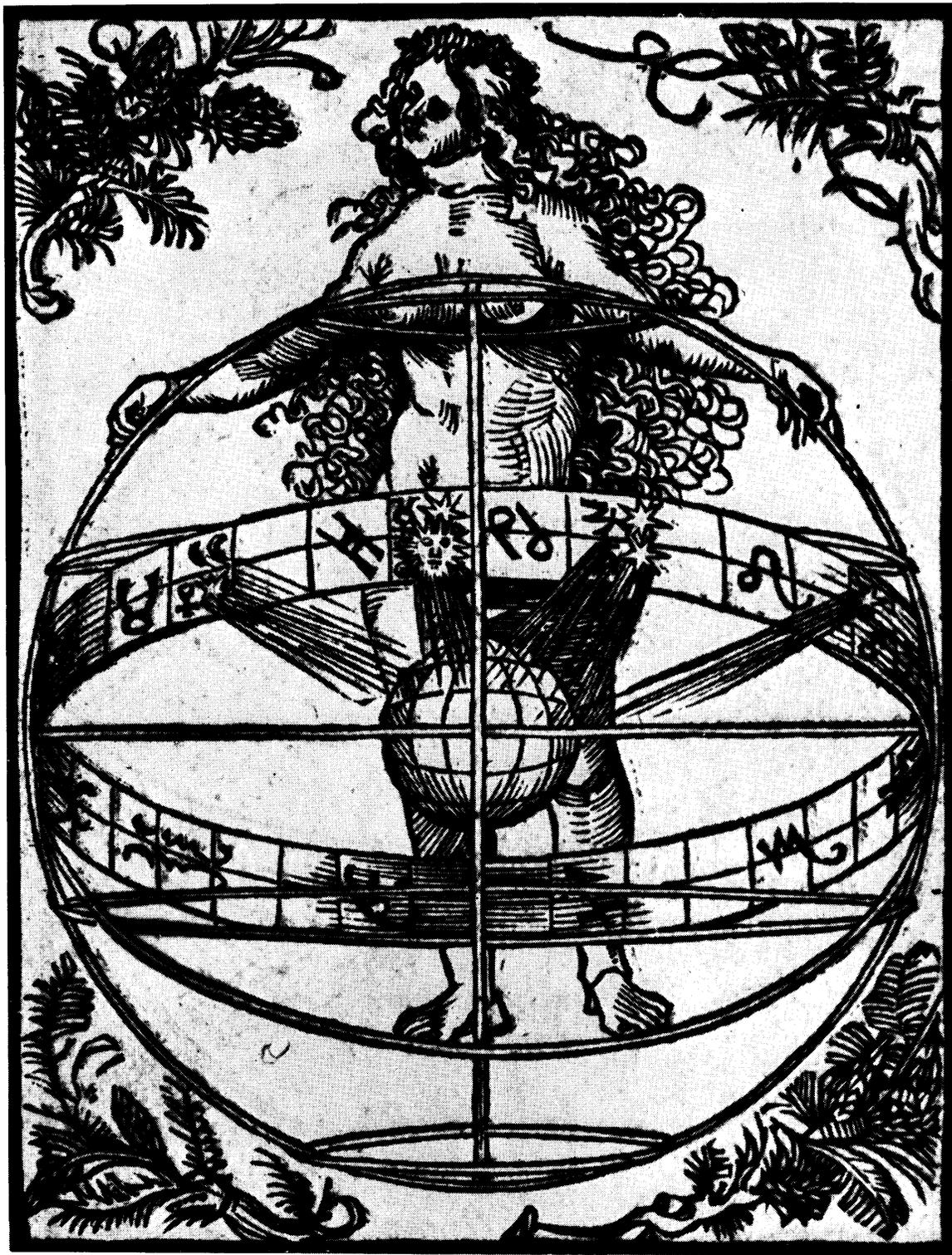
Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203

TATE 276H
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

The Compass

International NorthStar Users Association

Volume IV No. 6



INSUA News

It's the end of another good year for INSUA--we've held our own against the odds, and with the delivery of this issue have fulfilled our promise to get out six issues of Compass--well, yes, by the skin of our teeth, but we did it.

All 1984-85 members of INSUA should already have received notification of renewal and the plans for the annual meeting by first class mail; nevertheless, we will repeat the information here.

Membership Renewal

Membership in INSUA continues at \$20.00 per year. Once again, it will be the aim of INSUA to deliver six issues of Compass plus at least one freebie. The membership renewal offering this time around is a free disk. The tentative plan is for a floppy with utility programs on both sides, one side for CP/M users, the other for DOS users. It's premature to announce the precise programs to be distributed since they're still being tested. However, we promise they'll be useful to most INSUA members.

Annual Meeting

The annual meeting of INSUA will be held as usual during the West Coast Computer Faire in San Francisco. This year, as a departure from normal practice, the Faire will begin on a Saturday (March 30) and end on the following Tuesday (April 2). The Board of INSUA feels that Saturday afternoon, because it is not a weekday, is the only viable time to hold the meeting for maximum attendance. Please note that this will be the first day of the Faire! Another exception this time around is that INSUA will be holding the meeting in a hotel near the convention center rather than in the convention center itself. The reason for this is that the Faire management has raised the prices of regular booths to such a level that INSUA cannot hope to recover the costs by new memberships taken in at the booth. (The cost would run somewhere around \$3,000.)

We hope once again to have a talk from a representative of North Star, possibly with a presentation of the new Dimension computer.

Annual Meeting

Saturday, March 30, 1985

1:30-4:00 p.m.

San Francisco Computer Faire

Exact location to be announced

Projected speaker: North Star Representative

Announcements from the Chairman

Concluding with annual business meeting

and

Election of new officers

The Compass

The Compass is published every two months by INSUA, the Interational North Star Users Association, P. O. Box 2910, Fairfield, CA 94533.

Entire contents Copyright 1984 by INSUA. All rights reserved. Reproduction of material appearing in The Compass is forbidden without explicit permission. Send all requests to The Editor, Compass, P.O. Box 2910, Fairfield, CA 94533.

Subscription Information:

Subscription to The Compass is included in the \$20.00 INSUA membership fee. Address subscription inquiries to INSUA, P. O. Box 2910, Fairfield, CA 94533.

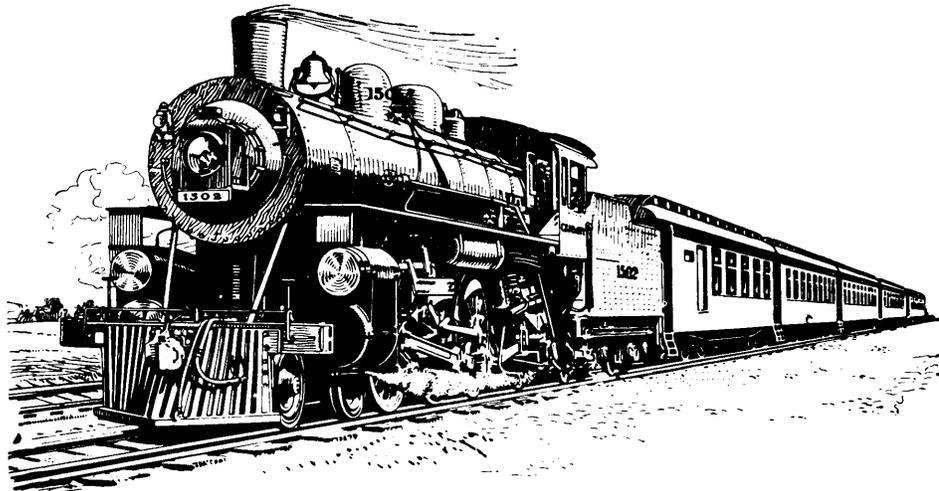
DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

Contents

- 2 Randy Fischer **TurboDOS and the Single User**
Description of TurboDOS, Functions, Features
- 4 David Allen **U. S. Robotics Modem**
Using & Installing USR Modem & MODEM7
- 11 John Shih **Semi-Disk Emulator**
Review of RAM Disk; comparison to floppies
- 12 Robert Floyd **Printer On??**
BASIC Program to test whether printer is on
- 13 Marianna van Erp **Multiple-line Headers**
Multiple-line Headers & Footers with WordStar
- 14 Perry Garst **String that Fyle**
Using files which are a long string
- 16 C. L. Makinson **Modification of Baud Rate**
A program to modify Advantage baud rate
- 19 Saul Levy **Relocating SECRETARY**
Description of N*'s MOVER, moving SECRETARY
- 27 Alan H. Nelson **Banking on your Computer**
Description of BofA's HomeBanking
- 31 Warren Lambert **Graphics**
SAS Typewriter Graphics for Program Evaluation
- 32 **Index of Volume IV**



TurboDOS and the Single User

By Randy Fischer
Fischer Computer Systems
445 Bay Street
Angwin, CA 94508

When most people think of TurboDOS they think of Networks and Multi-user systems. I would like to try to describe the benefits TurboDOS offers to the single user.

TurboDOS is an operating system written by a company named Software 2000. North Star Computers Inc. has been supplying TurboDOS for some time to their users. It was written originally for the Z-80, and with release 1.3 now supports the 8088/8086 microprocessors. It is designed to be a CP/M-compatible multi-user networking operating system.

The North Star implementation of TurboDOS supports the North Star Horizon computer and its related peripherals. There are several products for the Horizon Computer which were developed specifically to support TurboDOS. Since TurboDOS supports multi-user applications through networking several computers together with one computer as the Master (N* calls the Master a Server), and other computers as Slaves, North Star developed two new products, the UP-8 and UP-16. The UP-8 is a single-board Z-80 computer with 64K of RAM and two serial ports. The UP-16 is a single board computer which contains 128K of memory, an 8088 microprocessor, and two serial ports. There also is an optional 384K expansion memory board for the 8088 which gives a total of 512K of RAM to a UP-16 user. Both of these boards are designed to plug into the Horizon and are accessed through I/O ports by the existing Z-80 computer. By putting one or more slave boards, either Up-8's or Up-16's, into a Horizon, you have the hardware necessary for a TurboDOS System.

So a North Star TurboDOS system consists of a server computer which is a Horizon computer with at least one double-sided double-density floppy disk drive, an HRAM 64K memory board, and at least one of the hard disk systems that North Star supports. While it is possible to run TurboDOS on a floppy only system,

there is so much to be gained by using a hard disk that very few are using a floppy only system. Inside the Server are added the appropriate number of slave computers, either UP-8 or UP-16, for the particular application. In the North Star TurboDOS System there is a practical limit of 8 slave boards because of the number of S-100 slots available inside the Horizon. The TurboDOS operating system is used to give a CP/M-compatible system which shares the hard disk, floppy disks, and printers with the slave computers. TurboDOS also goes beyond CP/M and defines multi-user file access and other things needed for a good multi-user system.

Features

Now for some of the features that TurboDOS gives the user compared to what is available with the existing North Star implementation of CP/M:

****Print Spooling.** Character output can be placed on the hard disk and not sent to the printer until later. This allows you to generate the output of a program much faster than if you had to wait for the printer to finish before you could go on. At some time later you can send the file that was created for your program's data to the printer, which can go on while you are doing something else.

****Type-Ahead.** This allows you go enter data while the system is doing something else. For instance, while WordStar is loading, you can give it the commands to open a new file for editing etc., thus speeding up the process.

****Thirty-two User Areas.** This is twice as many as CP/M, very nice on a hard disk because you can better organize your data using a different user area for each different application on your system.

****File Attributes.** It is possible to set a

global attribute so files in user area zero can be accessed from any user area or drive. There is also an archive attribute which is set whenever a file is opened and reset by the COPY program--great for file backup of the hard disk.

Utilities

North Star's TurboDos comes with turbo plus, which supports a LOCATE command which will list all of the files on a drive sorted by user area. This is very nice for finding a file out there somewhere. There is no warm boot as with CP/M, so the system tracks do not have to be on any disk except the boot disk. There are also no built-in commands, so all commands come off the disk and thus are usually more powerful than the similar functions in CP/M:

** The RENAME command will allow ambiguous file references and can be directed to ask you for confirmation before changing a name.

** DELETE is nicer because of the YES/NO option before a given file is deleted.

** COPY can be used to copy all of the files on a disk with a YES/NO response for each file.

** The BUFFERS command will allow you to allocate the amount of memory to use as disk buffer space. Using this command allows you to assign large buffers for small, disk intensive programs, and less buffer space for large, non disk-intensive programs.

** The PRINTER command controls the printer assignments in the system. Files that have been spooled onto the disk can be assigned to different printers using this command.

** AUTOLOAD allows a user to set up commands which can be saved as files which execute either at initial load or at every warm start.

Configurations

The relative ease with which one can change the configuration of the TurboDos system is another of its advantages. All of the modules which make up a TurboDos

system are supplied with the operating system along with the linker to enable one to regenerate the operating system at any time. By using this ability, an experienced user can configure the system for specific needs. For instance, if you are not going to use printer spooling, that module can be left out of the operating system, thus saving more memory for other things. Also if a different disk controller is desired, the driver for that controller can be easily incorporated into the operating system. This is done simply by adding to or deleting from a text file called OSSERVER.GEN and adding the .REL files to the system.

There is also an OSSERVER.PAR file which contains patches to existing modules for specific users needs. The OSSERVER.GEN and PAR files are for the Server operating system. There are OSUSER-N.GEN and PAR files for the slave operating systems.

For the Single User!

All of the things described so far can be used to show how wonderful the TurboDos operating system is. But how can that benefit a single user?

The very things that make TurboDos a great multi-user system can be very useful for the single user. The ability to have multiple processors available to a single user opens up great possibilities. For instance, one can set up the system to use an editor on one slave computer while compiling or assembling a program on another. Through Batch processing the user can set up a sequence of instructions to another processor and have the task done while he is doing something else. Since TurboDos can also support the 16-bit processors, one can set up a system where you have easy access to either an 8-bit or 16-bit processor. One other nice thing is when you are ready for a multi-user system, you have simply to add a slave board for each new user.

Aside from being able to use the 8088 processor in the TurboDos system, the next biggest advantage is the ability to support Bank Switching on the Z-80 computers. While North Star itself does not currently support bank switching, there are relatively straightforward modifications that can be made to existing North Star hardware which will allow bank switching. With these modifications and the associated changes to the TurboDos

operating system, a TPA of 62.7K is possible. This makes for the biggest TPA available in any North Star Z-80 computer. This one thing alone can justify a TurboDos system.

If you need large amounts of memory,

8088 microprocessor capability, large hard disks, different floppy controllers, or the flexibility to support multi-user, then you should look very closely at TurboDos for your North Star Horizon.

#

The U.S. Robotics Modem and MODEM712

David M. Allen
Department of Statistics
University of Kentucky
Lexington, KY 40506

The U. S. Robotics S100 Modem is an excellent modem. It is an internal modem that gets its power from the mother board. There are no power supplies, electric cords, or boxes to clutter your work area. With an internal modem, the Horizon's second serial port remains available for use with a printer. Of course internal modems cannot easily be used with more than one computer.

Once the U. S. Robotics S100 modem is initialized, the commands that control it are a subset of the commands used for the Hayes Smartmodem. The U. S. Robotics Modem is not nearly as expensive as the Hayes Smartmodem. I have used mine for several months and am completely satisfied with it. Having used an acoustical coupler at 300 baud for some time, I found 1200 baud with automatic dialing like being in a new world.

Software is required to use any modem. U. S. Robotics sells a package called TELPAC for use with its modems. TELPAC is a nice package and is available in NorthStar CP/M format. However, MODEM 712, that came as a bonus with our 1984 INSUA membership, does most of the things TELPAC can do. (I think MODEM 712 alone is worth the membership fee to INSUA.)

MODEM 712 is structured so that everything that is specific to a particular computer, terminal, and modem is located at the beginning of the program. Assembly language source code, called an overlay, is provided for this first part of program. You modify this overlay to work with your equipment, assemble it, and then use DDT to place your equipment-specific code in the first part of MODEM 712. The DDT portion of this process is called installation. Except for how to modify the overlay, all of this is described in detail by comments in the overlay. The purpose of this article is show how to modify an existing overlay to use with the U. S. Robotics S100 modem.

INSUA's version of MODEM 712 comes with several overlays that represent the most commonly used NorthStar configurations. When I got my modem, I installed the Hayes Smartmodem overlay (M712HA.ASM). I did this because I had heard that the U. S. Robotics Modems were Hayes compatible. The resulting program caused Norton, my Horizon, to hang up. It turns out that the two modems are compatible at the "user" level but not at the electronic level. The initializations required by the two modems are quite different. It also turns out that the U. S. Robotics modem uses a single 8251 USART which is exactly the same USART used in the standard Horizon serial port. Consequently, the overlay supplied for use with the standard Horizon (M7NH-2.ASM) is a good starting point for producing an overlay to work with the U. S. Robotics modem.

Before using the modem one must set its switches to a port number not presently used. I used 80H. Use PIP to make a copy of M7NH-2.ASM called ROBOT.ASM. We will modify ROBOT.ASM to be our overlay. Begin by changing the port equate statement to match the modem port setting. I used

```
PORT: EQU 80H
```

CP/M initializes the standard serial ports upon power-up but not the modem. We need to add the following statements at the label INITMOD:

```
INITMOD: MVI A,0          ;CLEAR BYTE
          OUT MODCTL1
          OUT MODCTL1
          OUT MODCTL1
          MVI A,40H        ;RESET BYTE 01000000
          OUT MODCTL1
          LDA MODEBYTE     ;MODE BYTE
          OUT MODCTL1
          MVI A,37H        ;COMMAND BYTE 00110111
          OUT MODCTL1
          RET
MODEBYTE: DB 7AH
```

See the U. S. Robotics manual for a discussion of how to choose a modebyte for the desired baud rate, number of data and stop bits, and parity setting. If you want to use a printer attached to the second serial port while using the modem, make sure the LSTTST equate is

```
LSTTST: DB YES
```

You can then toggle the printer off and on using (CTRL-)P. Actually, this is all that is needed to have a useful working version of MODEM 712. However, check the rest of the article for some nice features that can be added.

To download COM files from CP/M bulletin boards one must use eight data bits and no parity. The University of Kentucky's IBM 3083 requires 7 data bits and even parity. Compuserve charges lower rates for 300 baud service than for 1200 baud service. So you may sometimes want to use 300 baud. You could have different versions, each with a different mode byte appropriate for a specific situation. It is much more convenient to modify the overlay so the SET command can be used to change the mode byte once the program is running. To do this, change the SETUPPTST equate to

```
SETUPTST: DB YES
```

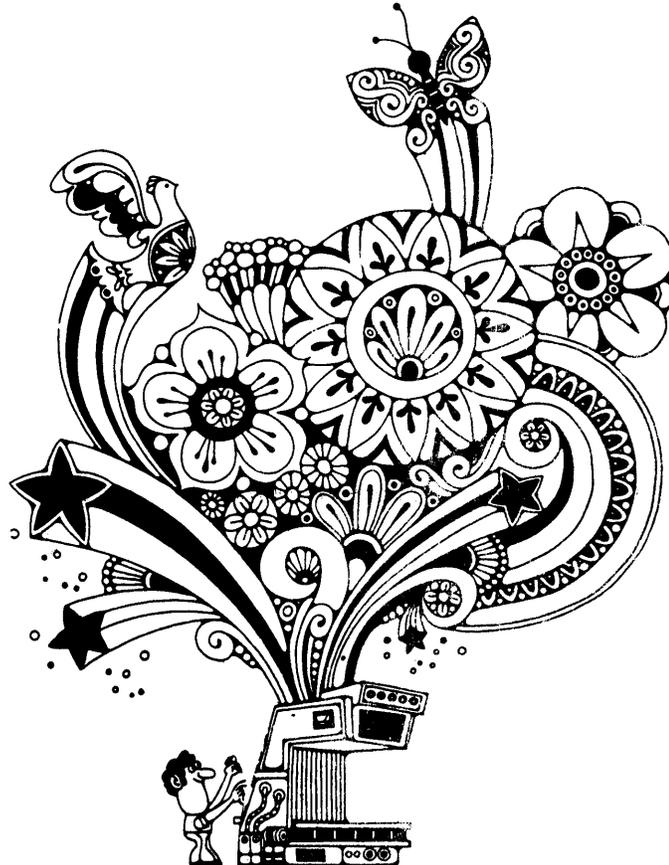
and at the label SETUPR add the code

```
SETUPR: LXI D,BAUBUF
        CALL JMP$ILPRT
        DB CR,LF,'1: 300 baud, 7 data bits, even parity'
        DB CR,LF,'2: 300 baud, 8 data bits, no parity'
        DB CR,LF,'3: 1200 baud, 7 data bits, even parity'
        DB CR,LF,'4: 1200 baud, 8 data bits, no parity'
        DB CR,LF,'Select: ',0
        CALL JMP$INBUF
        LXI D,BAUBUF+2
        CALL JMP$INLNCOMP
        DB '1',0
        JNC OK1
        CALL JMP$INLNCOMP
```

```

DB '2',0
JNC OK2
CALL JMP$INLNCOMP
DB '3',0
JNC OK3
CALL JMP$INLNCOMP
DB '4',0
JNC OK4
JMP SETUPR
OK1 MVI A,7BH
  STA MODEBYTE
  MVI A,1
  STA MSPEED
  JMP INITMOD
OK2 MVI A,4FH
  STA MODEBYTE
  MVI A,1
  STA MSPEED
  JMP INITMOD
OK3 MVI A,7AH
  STA MODEBYTE
  MVI A,5
  STA MSPEED
  JMP INITMOD
OK4 MVI A,4EH
  STA MODEBYTE
  MVI A,5
  STA MSPEED
  JMP INITMOD
;
BAUDBUF: DB 10,0
         DS 10

```



This takes care of the four situations I sometimes need. You may add or delete specifications for your own needs.

The next really neat thing to add is automatic dialing. Look at the file of phone numbers (M712NM.ASM). You will find the names and numbers of some of the gurus of microcomputing. Make a copy of this file called NUMBERS.ASM and change the names and numbers to those of systems you frequently call. In ROBOT.ASM change the SMARTMODEM equate to

```
SMARTMODEM: DB YES
```

If you have a touchtone phone you will also want to change the TOUCHPULSE equate to

```
TOUCHPULSE: DB 'P'
```

After installation you may auto-dial using the CAL command.

When I was on sabbatical leave at Pennsylvania State University, I used an editor that required a break to terminate insert mode. To have this capability, add the following code at the label SENDBRK:

```

SENDBRK: MVI A,08H ;send a break tone for 300 ms.
         JMP GOODBYE1
;.....
GOODBYE: MVI A,0AH ;set break, DTR low
;
GOODBYE1: OUT MODCTL1 ;put command register out of mode

```

```

IN MODCTL1 ;make sure it is now clear
IN MODCTL1 ;try once more
MVI B,3 ;delay for 300 ms.
CALL JMP$TIMER
MVI A,37H ;reset RTS, flags, DTR low, enable R/T
OUT MODCTL1 ;send to command register
IN MODCTL1 ;clear any incoming chars.
IN MODCTL1 ;try once more
XRA A ;clear the 'A' reg.
RET

```

The preceding code sends a 300 ms. break tone and sets DTR low for the same length of time.

My last addition is pure fluff. It causes a clear screen and a message when you first run MODEM 712. It also causes a clear screen between pages of the command menu. Change the SCRNTEST equate to

```
SCRNTEST: EQU YES
```

Enter the following code at the label CLREOS:

```

CLREOS: CALL JMP$IILPRT ; 195H
        DB 1BH,'Y',0,0,0 ; 198H
        RET ; 19DH
;
CLRSCRN: CALL JMP$IILPRT ; 19EH
        DB 1AH,0,0,0,0 ; 1A1H
        RET ; 1A6H
;
SYSVER: CALL CLRSCRN ; 1A78
        CALL JMP$IILPRT ;
        DB CR,LF,'NorthStar Horizon Computer'
        DB CR,LF,'U. S. Robotics S-100 Modem'
        DB CR,LF,'TeleVideo 912 or 920 Terminal'
        DB CR,LF,0
        RET

```



CLREOS stands for clear to end of screen, and CLRSCRN stands for clear screen. The bytes used here are specific for the TeleVideo 912 or 920 terminals. These must be changed to the corresponding values for your terminal. The number of bytes in the DB statements must remain five. Of course the message regarding the terminal should also be changed.

After you have made the modifications to ROBOT.ASM that you want, assemble and install it. If you want automatic dialing, assemble and install NUMBERS.ASM by following the example in the comments. I have found the resulting program to be very useful. For those of you who don't want to bother punching this in, INSUA has my permission to include the full ROBOT.ASM in their next software distribution. If you don't want to wait, send me \$10.00 and I will send you a copy within a week.

#

Semi-Disk Emulator

By John Shih

(The following article is reprinted from the January 1985 issue of Polaris, the Newsletter of the North Star Computer Society, Seattle Washington. -Ed.)

SemiDisk Systems makes a S-100 board which provides disk emulation using RAM chips and can be used in the North Star Horizon. I purchased my SemiDisk about a year ago with 32 64K x 1 bit dynamic RAM chips installed for a capacity of 256k bytes of memory. I later expanded the total memory to 512K bytes of additional 32 64K chips in the sockets provided. Several SemiDisk boards can be used together to obtain even greater RAM disk capacity. A different version of the battery back-up option is also available to protect the data in case of power loss to the computer.

The SemiDisk board is accessed through any four contiguous port addresses selectable by dip switches. SemiDisk provides a self-installing driver in the form of a CP/M COM file, which locates itself under the CCP and changes the BIOS jump table to allow access of the RAM disk. The driver takes about 750 bytes.

Since I purchased SemiDisk, I've received a free upgrade (version 5.0), which can be located above the Horizon ROM memory window. It also allows a portion of the SemiDisk to be used as a spooler for printer buffering. Source code for the driver is provided, but Digital Research's MAC assembler is required for assembly. To enable the spooler option requires re-assembly. As I don't have MAC, I have tested the spooler option.

To bring SemiDisk up on a Horizon is very easy. Just plug the SemiDisk board into the computer, boot the computer, install the driver by typing SEMIDISK.COM and you have an extremely fast disk drive in memory, which acts almost as any other disk drive. The self-installing driver has a number of options. It will modify itself to use the proper port addresses once you've initially entered them. You can also configure the RAM disk to be any drive allowable by CP/M. If you already have a disk drive assigned to a particular letter that you want to use for the RAM disk, the original drive is assigned to the next higher letter. For example; if you have a

floppy disk designated as Drive B: and you assign SemiDisk to Drive B:, then the floppy will be moved to C:. SemiDisk cannot be partitioned into more than one logical drive. However, different user areas can be set up as usual.

I've found SemiDisk to be extremely useful. Programs which require a lot of disk access, such as CBASIC compiler and many dBase II operations, run much faster. Equally important; these programs now run in blissful silence instead of the constant noise of disk drives grinding away.

To provide some idea of the speed advantage, using the Rev 4.1 of SemiDisk, I have compiled a 950-line CBASIC program and sorted a 211-record dBase II file in five separate test situations; 1) a DD/DS floppy drive, 2) a 15MB hard disk, 3) Semidisk, 4) programs on the hard disk with the data files on SemiDisk. The CBASIC compilations were run with the \$GB option, which wrote the listing to a disk file instead of to the console. The results are given in Fig. 1.

The speed advantage of SemiDisk is quite obvious. However, as it is a RAM disk, when power to the computer is lost, so are the contents of the SemiDisk, unless the battery back-up option is installed. Therefore, to prevent loss of data all files on the SemiDisk must be copied to floppy or hard disk after each work session in which any updates have been made. I've developed the strategy of leaving the application programs (e.g. dBase II .CMD files) on the hard disk and writing the data files to SemiDisk at the beginning of a work session. After I have finished with work files, I use PIP to copy the files that I care to save back onto the hard disk. If I had to choose between a hard disk and a RAM disk, I would choose the latter, as I am frustrated when waiting for an application program to complete what it is doing.

In addition to the danger of loss of power, which has happened twice in the last year in West Seattle during business

hours, an incompatibility exists between the SemiDisk driver and the Spellbinder word processor that I use. CP/M does not warm boot properly after exiting from Spellbinder. Spellbinder works fine until I attempt to exit to CP/M. I have to then cold-boot to exit Spellbinder. I have not encountered this problem with any other software package, including Supercalc, CBASIC, dBase II and Turbo Pascal. Fortunately, the data on SemiDisk is unaffected after a cold-boot and re-installation of the SemiDisk driver works fine as long as power has not been turned off.

Programs that are very demanding on memory space may have problems using SemiDisk. Our Horizon uses North Star's CP/M Rev. 2.2 with a full 64KB of memory. Parts of BIOS are located above the North Star ROM window. Even so, the shortest

JRT Pascal (Rev 3.0) program does not compile due to insufficient memory when the SemiDisk driver is loaded. However, I've given up on JRT Pascal, since receiving Turbo Pascal.

In summary, if you use your Horizon with programs that do a lot of accessing of the disk, then you should find SemiDisk helpful. I would even rank it above a hard disk for usefulness. It is a well-designed product that is easy to install and operate.

(The editor of Polaris points out that a hard disk "serves another very important function that neither the RAM disk or floppy can accomplish, and that is storage of a large volume of data several orders of magnitude above that of the largest current floppy drive. -Ed.)

#

CBASIC Compilation dBase II Sort

Floppy Disk	5 min. 47 sec. (100%)	1 min. 39 sec. (100%)
Hard Disk	3 min. 24 sec. (59%)	1 min. 04 sec. (65%)
Floppy/SemiDisk	2 min. 29 sec. (43%)	36 sec. (36%)
Hard/SemiDisk	2 min. 23 sec. (41%)	36 sec.
SemiDisk	2 min. 23 sec. (41%)	34 sec. (34%)

Fig. 1

Printer On??

By Robert Floyd

The following routine allows testing from program flow to see whether the printer is turned on.

```

8000 REM ** The following function selects the print location on the screen **
8010 REM ** of my Televideo 912C terminal - may vary for others.          **
8020 REM
8030 DEF FNT$(X,Y)=CHR$(27)+CHR$(61)+CHR$(32+X)+CHR$(32+Y)
8040 REM -----
8050 REM ** The following string clears the screen of my terminal.      **
8060 REM
8070 C#=CHR$(27)+CHR$(43)
8080 REM -----
8090 REM ** The following subroutine checks to see if printer is turned on. **
8100 REM ** On my Okidata Model 92 printer, if the printer is on a 51 is   **
8110 REM ** returned and if the printer is off a 59 is returned. This may **
8120 REM ** be tested by typing "PRINT INP(6)" with printer off and with   **
8130 REM ** printer on. These tests are for the parallel port but no doubt **
8140 REM ** something similar will work with the serial port.           **
8150 REM
8160 IF INP(6)=51 THEN RETURN
8170 PRINT CHR$(7)
8180 PRINT C#, FNT$(10,22), "PRINTER IS NOT READY"
8190 FOR X7=1 TO 1200 \ NEXT \ REM ** Delay **
8200 GOTO 8160

```

Multiple-Line Headers

By Marianna van Erp

(Editor's note: Way back in Vol. II, no. 2 of Compass, Alan Lewis Painter asked whether WordStar could produce multiple-line headers. The following solution to this problem was recently published in Foghorn, an Osborne Users' Newsletter which has now expanded to include other users as well. We are grateful to the editors of Foghorn and Marianna van Erp for permission to reprint the article.)

WordStar normally provides one-line headers and footers. The following imaginative procedure, which appeared in the "*" column in PC World, allows up to 200 characters in headers and footers, with as many lines as you like.

Part 1) tells how to make a new control, (Ctrl-)R, which sends a line feed and a carriage return to the printer. Part 2) tells how to type a multiple-line heading. Part 3) explains the procedure.

PART 1: Making the (Ctrl-)R control

Use a COPY of WS.COM. Define User Patch #4 to make (Ctrl-)R send the printer a carriage return and a line feed. (You may use any of the four user-defined functions that WordStar has vacant; #4 is used in this discussion.)

For WordStar 3.3, run WINSTALL. From the installation menu, choose Custom Installation of Printers; from that menu, choose Item Q, User Defined Functions. Go to Function #4. You will be asked to enter new values. Enter 0A 0D 00 00, i.e. two characters in each of the four places to enter values.

For earlier versions of WordStar, use DEBUG. Put a copy of DEBUG.COM onto the same disk as WS.COM. Type on the A-prompt:

```
DEBUG WS.COM
-F 78E L3 02 0A 0D (CR)
-W (CR)
-Q (CR)
```

(Editor's note: We don't know DEBUG.COM; however, older versions of WordStar have an INSTALL.COM which allows the same kind of custom installation as the new WINSTALL. See your WordStar manual, or the various WordStar and Epson Printer articles in Compass, Vol. II, no. 3.)

Part 2: Creating a multiple line heading.

Put the cursor in column 1. Use (Ctrl-)0X to release the right margin. Type:

```
.HE FIRSTLINE(Ctrl-)R
↳(8 spaces)SECONDLINE(Ctrl-)R
↳(8 spaces)THIRDLINE
```

(Editor's note: The three lines above represent one line on your screen. To place a (Ctrl-)R in the line, type (Ctrl-)P followed immediately by R or (Ctrl-R.)

Type (Ctrl-)0X again to "un-release" the margin. Put the cursor on the dot of the .HE and hit (Ctrl-)N to open up a line at the top of the file. On this top line type in the dot command:

```
.pl n
```

The number n is 66 MINUS the EXTRA lines of the heading.

Warning: Do not use print-controls for dark (double-strike) print or bold print in the heading. Multiple printing of the heading with line-overlaps will result, a logical result of WordStar's thinking of our two or three lines as one line.

PART 3: Explanations

Why the 8 spaces between each line of the heading? A (Ctrl-)R in the file will send the printhead down one line and all the way to column 1 of the paper. The 8 typed spaces in the heading send the printhead over to column 8, which is WordStar's default first print column. If you use a different left margin, adjust the

number of spaces in the heading.

Why the .PL (Page Length) command? WordStar does not "know" about the extra lines in the heading. We are fooling it, but it will want to retaliate by making each page that many lines longer; the top line of the following pages will start lower and lower on each successive page. WordStar "thinks" the page should be 66 lines long. We fool it once again by

telling it that the page length (.PL) is 65 lines for a 2 line heading or 64 for a 3-line heading (66 minus the extra heading lines).

This same technique may be used to create multiple-line footers. The default bottom margin of 8 lines is ample to carry an extra line. A page may have both multiple line headers and footers. Use a .PL of 66 minus ALL extra lines.

#

String that Fyle

By Perry Garst
Rt.1, Box 1295
Quinton, OK 74561

North Star BASIC strings differ from other BASIC strings in several ways. Substrings are accessed by their position in the string without the additional "LEFT\$,MID\$,RIGHT\$" notations. Thus if A\$="department", A\$(3,6)="part". The string can be corrected or changed by changing the substring. A\$(4,4)="o" changes A\$ to "deportment". Other BASICs may require separating the string into parts, making the change, then concatenating. Instead of just printable characters, North Star allows any of the 256 values of a byte to be used as a character. A big difference is that North Star string length is "limited" to 65536 characters rather than to the more common 256. There is no way to get a string of 65536 characters into a standard system along with DOS and BASIC, but it does allow very long strings to be used.

A string has some similarities to a disc file. Substrings can be set up to give sequential or random access to parts of the string. If a string is constructed as a series of records, the string itself becomes a file. It can be stored to and retrieved from a disc as a string but it is accessed and manipulated in memory and not from the disc. It is just as easy to get a single record from a string as it would be from a disc file and it is faster. As long as the string will fit in memory, a file can be written as a single string.

One difference in using a string instead of a normal file is that all the spaces in a record must be filled. In a normal file this is not necessary because the various parts of the record are accessed by overhead bytes that tell BASIC what and how long that segment is. The only overhead bytes in a string file are in front of the string to tell how long the string is. This is an advantage for compactness because the overhead bytes are eliminated, but it means that each field in a record must be exactly the same length and in the same position. If 20 spaces are allotted to "Description", for instance, "TELEPHONE" would have to be extended to 20 characters by adding nine characters (e.g. blanks) to it. Numbers are another problem. The available "STR\$" and "VAL" are not easy to confine to an allotted space and use more characters than is necessary.

Here is where the 256 values of a string character can be used. A single character can represent a number between 0 and 255. Two characters can represent 65536 numbers (256X256) and three characters, any number between 0 and 16777215 (256X256X256). Depending on the range, a number field can be one or more characters and the calculation will fill the allotted space. These numbers are integers, so decimal fractions will have to be handled in the program.

The accompanying program demonstrates how a string can be used as a file. Each record is 25 characters long. "CATEGORY" is a single character as a number, "AMOUNT" is a three character number to represent dollars and cents. "DESCRIPTION" is a 21 character string which is filled out with spaces if it is short. User definitions FNA(N2) and FNB(N3) convert "AMOUNT" to characters and back to numbers using CHR\$ and ASC. You can enter up to 20 records if you answer "Y" to "CONTINUE?". A\$ is then put in disc file "DSCFILE" which is a two-block data file that you must create (i.e. CREATE "DSCFILE",2). A\$ is then nulled and written on the screen to show that it is empty. DSCFILE is READ for A\$ which is then printed sequentially. Random access is demonstrated after that.

There are, of course, many things that

could be done to make this a useful program. For a permanent file, A\$ would be read from the file first and then added to, not started each time the program is run. The category could be printed as a word keyed by the number, more fields could be added, etc. 500 characters is a small string considering the memory in most North Stars today. The string can be much larger. On the other hand, this method of constructing a file is compact, so a small string can contain a lot of information. Next time you need a file in your program, try a string file.

Author's note: This is a sequel to "Byte That Fyle", Compass, Vol. III, no. 3. It was correctly pointed out by Saul Levy, Compass, Vol. III, no. 4, that byte access to a disc is painfully slow. So it was back to the keyboard, and this is the result.

#

```

100 DIM A$(500),B$(21)\A$=""
110 INPUT"CATEGORY ",N\B$=CHR$(N)\A$=A$+B$
120 INPUT"AMOUNT ",N1\N2=FNA(N2)\A$=A$+B$
130 INPUT"DESCRIPTION ",B$
140 IF LEN(B$)=21 THEN 150\B$=B$+" \GOTO 140\REM FILL B$
150 A$=A$+B$\IF LEN(A$)=500 THEN 170
160 \INPUT"CONTINUE? ",Y$\IF Y$="Y"THEN 110
170 OPEN#1,"DSCFILE"\WRITE#1,A$\CLOSE#1\REM PUT A$ ON DISC
180 A$=""\FOR X=1 TO 24\A$\NEXT\REM EMPTY A$ AND PROVE IT
190 OPEN#1,"DSCFILE"\READ#1,A$\CLOSE#1\REM GET A$ BACK FROM DISC
200 !"RECORD NO.",TAB(15),"CATEGORY",TAB(30),"AMOUNT",TAB(50),"DESCRIPTION"
210 \FOR X=0 TO LEN(A$)/25-1\GOSUB 240\NEXT\REM SEQUENTIAL ACCESS
220 INPUT"ENTER RECORD NUMBER ",X1\IF X1>LEN(A$)/25 THEN 220\REM RANDOM
230 X=X1-1\GOSUB 240\INPUT"CONTINUE? ",Y$\IF Y$="Y"THEN 220 ELSE END
240 A=ASC(A$(X*25+1,X*25+1))\N3=FNB(N3)\B$=A$(X*25+5,X*25+25)
250 !X+1,TAB(15),A,TAB(30),%9F2,N3,TAB(50),B$\RETURN
260 DEF FNA(N2)\REM CONVERT NUMBER TO THREE CHARACTER STRING
270 N1=N1*100\A=INT(N1/65536)\B=INT((N1-A*65536)/256)\C=N1-A*65536-B*256
280 B$=CHR$(A)+CHR$(B)+CHR$(C)\RETURN(N2)
290 FNEND
300 DEF FNB(N3)\REM CONVERT STRING TO NUMBER
310 N3=ASC(A$(X*25+2,X*25+2))*65536+ASC(A$(X*25+3,X*25+3))*256
320 N3=N3+ASC(A$(X*25+4,X*25+4))\N3=N3/100\RETURN(N3)
330 FNEND

```

Modification of Advantage Baud Rate

By C.L. Makinson
TSR Microcomputer Consulting
PO Box 9191
Akron, OH 44305

Several methods exist for changing the baud rate on the North Star Advantage under CP/M. This article reviews two techniques which make a permanent change to the BIOS and one technique that is resident only until cold booting.

Where the computer will always be connected to a printer via serial port#1, the baud rate need only be set one time at system configuration. This is done by running CPMGEN and answering the prompt at Q6 with the appropriate baud rate for the printer.

An alternate method involves using CP/M's SYSGEN and DDT to patch the BIOS. By referring to your factory-supplied USER.ASM you can see the labels for BAUDA,B,C and PORTAB,BB,BC. These define the locations to patch for baud rate and port respectively. Note from the assembly remarks that BAUDA is for port#1 and BAUDB and BAUDC are for port#2, which defines both the CP/M reader and punch devices.

Figure 1 shows the procedure for making the patches using SYSGEN-DDT-SYSGEN. If more than 2 serial ports are installed or if port#2 must function as other than a reader/punch, then the factory-supplied USER.ASM must be modified. Modification of USER.ASM is beyond the scope of this article.

The final method to be discussed is a temporary baud rate change effected by issuing an OUT instruction to the appropriate port address with the appropriate baud rate code as illustrated in Table 1. This can be done from any language with an OUT command or with an assembly program like the ADBAUD utility in Listing 1.

But why a temporary change? A example of where this would be of use would be in transferring files between an Advantage and an alien computer using MODEM7 and direct connected cables. In order to avoid one hour per disk transfer time, the baud rate is set well above the normal 300 baud modem rate.

Unfortunately, because of synchronization problems, few alien computers can communicate with the Advantage at 9600 baud using MODEM7, and I know of none that work at 19200. It is often the case that receive link can be 4800-9600, but when sending, the Advantage must be slowed to 1200-2400 to allow the alien computer to keep up. The maximum successful rates are determined by trial using ADBAUD to change the rate. This is much quicker than the BIOS patching technique.

A more extensive version of ADBAUD is available from the address above for \$10. This version includes a help screen, menu, will handle five serial ports, and can be run from the command line as follows:

```
A>ADBAUD baud port      syntax
A>ADBAUD 9600 3         set port#3 to 9600
A>ADBAUD 600           set default port#1 to 600
A>ADBAUD                display selection menu
```

#

PATCHING BAUD RATE WITH DDT
User input shown in < >.

```

A><SYSGEN> Load system to memory.
<1> Reply to source prompt (read from drive 1).
<^C> Reply to destination prompt (warm boot).
A><DDT> Patch the memory image.
-<S3218> Substitute values starting at 3218 hex.
3218 7E <78> Substitute 78 for 2400 baud at BAUDA.
3219 70 <40> Substitute 40 for 300 baud at BAUDB.
321A 70 <40> Substitute 40 for 300 baud at BAUDC.
321B CD <.> Quit substitute mode.
-<^C> Return to operating system.
A><SYSGEN> Copy modified system back to disk.
<CR> Reply to source prompt (already in ram).
<1> Reply to destination prompt (write to drive 1).

```

FIGURE 1

BAUD RATE TABLE

BAUD RATE	DECIMAL	HEX
=====	=====	===
19200	127	7F
9600	126	7E
4800	124	7C
2400	120	78
1200	112	70
600	96	60
300	64	40

I/O ADDRESS TABLE

BOARDLET SLOT DECIMAL					BOARDLET SLOT HEX					FUNCTION
1	2	3	4	5	1	2	3	4	5	
=====										
80	64	48	32	16	50	40	30	20	10	USART DATA
81	65	49	33	17	51	41	31	21	11	USART STATUS
88	72	56	40	24	58	48	38	28	18	BAUD RATE REGISTER

EXAMPLE TABLE USAGE: CHANGE PORT 1 BAUD RATE TO 2400

BASIC:
OUT 88,120

ASM:
MVI A,78
OUT 58

TABLE 1

PROGRAM ADBAUD
LISTING-1

```

; northstar advantage baud rate select
; rev: 1/08/85CL Makinson shortened for publication
; rev: 1/12/83 WT Prewitt
;
; sets baud rate of northstar advantage serial port no. 1
;
; assumes boardlet no.1 is present and is serial type and
; usart is initialized by bios
;
; (8080 code- use cpm assembler & loader as follows:
; asm adbaud
; load adbaud
;
; org 100h
; real places
; wboot: equ 0 ;jump here to warm boot
; bdos: equ 5 ;jump vector addr for bdos
;
; ascii equates
; cr: equ 0dh ;carriage return
; lf: equ 0ah ;line feed
; bell: equ 7 ;ctrl-g rings bell
; dollar: equ 24h ;dollar sign to terminate string
; cpm bdos functions for C reg
; pstr: equ 9 ;print string in de reg
; rdstr: equ 0ah ;read string into de regs buffer
;
;-----
start: lxi d,signon ;send signon msg
; call xmit ;send string
;
; loop: lxi d,menu ;menu
; call xmit
; call recv ;get char in a
; cpi '0' ;was it 0 for exit?
; jz wboot ;if so, quit
;
; do3: cpi '1' ;was it 1 for 300 baud
; mvi c,40h ;300 baud
; jz doit ;set it
;
; do6: cpi '2' ;was it 2 for 600 baud
; mvi c,60h ;600 baud
; jz doit ;set it
;
; do12: cpi '3' ;was it 3 for 1200 baud
; mvi c,70h ;1200 baud
; jz doit ;set it
;
; do24: cpi '4' ;was it 4 for 2400 baud
; mvi c,78h ;2400 baud
; jz doit ;set it
;
; do48: cpi '5' ; ;
; mvi c,7ch ;4800 baud
; jz doit ;
;
; cpi '6' ; ;
;
; do96: mvi c,7eh ;9600 baud
; jz doit ;
;
; do19: cpi '7' ;
; mvi c,7fh ;19200 baud
; jz doit ;
;
; jmp loop ;else start over
;-----
;
; doit: mov a,c ;new baud char
; port: out 58h ;to serial boardlet 1
;
; jmp wboot ;go home
;-----
;
; xmit: mvi c,pstr ;xmit string ident
; call bdos ;call cpm bdos
;
; call stall
; call stall
; ret
;-----
;
; recv: mvi c,rdstr ;read a string
; lxi d,rdbuf ;point to buffer
; mvi a,80 ;ridiculous max buffer size
; rdbuf ;max to buffer
; xra a ;zero
; sta rdbuf+2 ;wipe lst loc
; call bdos ;get string
; lxi d,rdbuf+2 ;point to lst char loc
; ldax d ;get first char
; ret
;-----
;
; stall: mvi b,0 ;set up time waster
; stalp: dcr b
; jnz stalp ;loop 255 times each
;
; stalp1: jnz stalp1
; stalp2: dcr b
; jnz stalp2
; ret
;-----
;
; signon: db 'Northstar Advantage Baud Rate Utility'
; db cr,lf
; db 'rev 1/12/1983 W.T.Prewitt'
; db cr,lf,dollar
;
; menu: db cr,lf,'Advantage Serial Port Baud Rate Select',cr,lf
; db '0= Exit to system without changing',cr,lf
; db '1= 300',cr,lf
; db '2= 600',cr,lf
; db '3= 1200',cr,lf
; db '4= 2400',cr,lf
; db '5= 4800',cr,lf
; db '6= 9600',cr,lf
; db '7= 19200',cr,lf
; db dollar
;
; rdbuf: ds 82 ;console read buffer
; ds 64 ;huge stack
;-----
; end

```

Relocating SECRETARY with North Star's MOVER

By Saul G. Levy
2555 E. Irvington Rd., Lot 47
Tucson, Arizona 85714
(602) 889-7753

Load SECRETARY at ANY Address

Yes, dear North Star user, you can now run SECRETARY with any Release 5 or later DOS, at any load address you want, on any Horizon or Advantage Computer! The version I am using to write this article loads at 1000H and runs fine with 2.1.1 DOS. 5.2 DOS also works because all DOS-address jumps are at the same locations (both DOSes load at 100H).

How Does MOVER Work?

Release 5.2 documentation contains a short description of MOVER (1 1/3 pages) which does not explain how MOVER works. Documentation for 2.1.1 has not been published (I am working on it!). MOVER is a very nice program which can also relocate non-North Star software.

The REMARKS at the beginning of MOVER describe what is needed to relocate program files:

1. A standard copy of each program (i.e. a running copy)
2. A copy of each program assembled for 0H
3. The DOS and boot PROM calls must also be assembled for 0H

To simplify what follows a bit, I will use Release 5.2 BASIC as our file to be relocated. Point 1 above refers to the file BASIC on the 5.2 system software diskette. Points 2 and 3 refer to the -BASIC file which the documentation calls the relocation key file (I call it the minus file).

The Minus File

Listing 1 is the beginning of the BASIC file loaded at its normal E00H address.

Note the jump in E01-E03H: C3 05 0E, which says JMP 0E05H. The load HL immediate in E05-E07H: 21 1D 42 says LXI H,421DH. Now look at the same code in Listing 2 which is from the -BASIC file. The jump is now to 0005H and the load immediate is now 341DH. The BASIC file was assembled for E00H; the -BASIC file for 0H. The difference is a constant 0EH in the high byte (0EH-0EH=0H and 42H-0EH=34H). The other bytes are identical in both versions and do not need to be relocated (a difference of 0H).

Now look at the E8H byte in E10H of BASIC. This is the high byte of the boot PROM located at E800H. Point 3 above says to zero (0) this byte which is what is found at E10H in the -BASIC file. Note that the difference is E8H, not 0EH (E8H-E8H=0H)! Finally, look at the calls in the bottom sections of the listings at 13B1-13B3H and 13EE-13F0H: CALL 011CH and CALL 0122H. These are DOS addresses which have a difference of 01H.

That is nice, but how does MOVER work? What MOVER does is to compare each byte in the BASIC and -BASIC files, determine the difference (if any), and then add an OFFSET (if a difference is found) to the byte from the minus file. The offset value depends on WHICH difference is found and WHERE you want to run DOS and BASIC (the differences are: 0EH for BASIC addresses, E8H for boot PROM addresses, and 01H for DOS addresses). You have to enter enough information to tell MOVER what each of the corresponding offsets are, then MOVER will merrily write the unchanged or offset bytes to the minus file which destroys the ability to relocate the minus file to any other location (as North Star warned: Always relocate a working copy of the files, never the master diskette itself!).

Using MOVER

Using MOVER is a bit complicated until you have used it a few times or understand what it is doing. Both versions ask the following questions:

1. Are you ready to proceed?
2. Will this software be used in a dual-density/quad computer?
2A. Will this software be used in a single-density computer?
3. Will this software be used in a computer with a nonstandard bootstrap PROM set?
3A. What is the origin for the PROM set (in hex)?
4. What is the origin for the new DOS (in hex)?
5. Should BASIC, utilities, etc. follow DOS?
5A. What is the preferred origin for these (in hex)?
6. Are all origins specified as intended?

Question 1 expects a Y (RETURN) or the program will stop. Question 2 or 2A will appear depending on which density the program thinks you are using (double is the default unless the single-density, Release 4, M2A00 Monitor is on the diskette!). A Y (RETURN) is expected, otherwise the density will be switched and the question repeated for the other density until you give a Y (RETURN)! This style of coding is questionable, but it does work.

Question 3 is unnecessary as long as you have a standard Horizon with the boot PROM at E800H. The oldest, single-density, disk controller boards used E900H (I have no idea why North Star changed this address by 100H). If you have an E900H, or nonstandard, or phantomed-out boot PROM, be sure to read the end of Using ASSYZERO below (starting at 1030..)! Question 3A will appear if you answer with the (usual!) Y (RETURN) to Question 3. 3A expects from one to four digits (E9 or E900, etc.).

Question 4 is asked only if the computer can use double density (North Star goofed on 5.2S, the single-density version of 5.2DQ. You had to buy Pavel Breder's Boot Package to relocate the 5.2S DOS!). Enter from one to four digits as for Question 3A (1 or 100 for the normal origin, anything else for a nonstandard origin). Note that all origins must be on a

page boundary of 100H (at 100H, 200H, 300H, etc., but not 101H, etc.). Also, no program can wrap around the end of memory which means that you cannot load the DOS at EC00H (above the boot PROM) and then load BASIC at F900H and continue loading from FFFFH through 0H and on up. This is standard assembly language practice (a few special programs do load at a non-page boundary). MOVER prints this information before Question 2 is asked.

Question 5 expects a Y (RETURN) only if all of the other programs will load at the normal address (E00H), otherwise Question 5A is asked and expects from one to four digits as for Questions 3A and 4. Question 6 expects a Y (RETURN) to start the relocation, otherwise it will return to Question 2 allowing you to start over again.

MOVER expects the disk directory to be labeled in one of the following ways:

CR <*> 8 0 For double density

or

CR <*> 4 0 For single density

then TYPE it with:

TY <*> 3

This makes the disk directory into a BASIC data file with the name<*>. The directory is opened and read one byte at a time. MOVER will relocate all files that start with a minus sign (the minus files). Each pair of these files (minus and regular) is opened, read four bytes at a time (to speed things up), relocated, and written to the minus file. Then the directory is updated with the new file TYPE and GO ADDRESS for the minus file. If the non-minus (plus?) file is not on the diskette, the program will stop on a FILE ERROR.

Understanding MOVER

MOVER is a very ingenious program. If you are not interested in a few more details, please skip to the last paragraph of this section, otherwise, print a copy of MOVER and follow along.

During and just after the question section described above, MOVER is storing

```

>DH E00,90
0E00 AF C3 05 0E 37 21 1D 42 11 FF 7F C3 21 0E 50 01
0E10 E8 83 3B 18 C3 DF 0E 5F 00 70 41 75 40 8A 3F 3D
0E20 3E 3E 01 32 26 3E 23 22 39 3E EB 22 3B 3E 3E 00
0E30 32 2C 3E 21 7A 3B F9 22 ED 3B CD D9 3A 3E 01 32
0E40 EB 3B 3E 00 32 7A 3B 32 E3 3B D2 A3 0E 2A 39 3E
0E50 7E FE 01 CA 80 0E B7 CA 68 0E CD 9B 36 EB 21 3B
0E60 3E CD CF 29 EB DA 50 0E 2A 39 3E 3A 2C 3E B7 CA
0E70 75 0E 2A 2F 3E 36 01 CD 06 13 01 1C 11 C3 F1 0F
0E80 22 37 3E 3A 2C 3E B7 CA 9D 0E 2A 2F 3E 23 CD B5
>

```

```

DH 13A0,60
13A0 E5 32 2C 3E 3E 01 32 26 3E CD D2 38 CD 8D 29 3E
13B0 01 CD 1C 01 DA A0 0F 4F CD B5 26 D5 23 23 7E E6
13C0 7F FE 02 C2 B8 0F 7E E6 80 B1 4F 23 7E B7 CA D8
13D0 0F E1 D1 D5 E5 F5 2A 3B 3E CD 29 17 F1 BC DA E2
13E0 13 7C 0C 0D F2 EA 13 C6 01 1F E1 D1 06 01 CD 22
13F0 01 DA D8 0F 37 C3 33 0E CD 8D 29 E5 E5 3E 01 CD
>

```

Listing 1: Dump of BASIC file

```

>DH E00,90
0E00 AF C3 05 00 37 21 1D 34 11 FF 7F C3 21 00 50 01
0E10 00 83 2D 18 C3 DF 00 5F 00 70 33 75 32 8A 31 3D
0E20 30 3E 01 32 26 30 23 22 39 30 EB 22 3B 30 3E 00
0E30 32 2C 30 21 7A 2D F9 22 ED 2D CD D9 2C 3E 01 32
0E40 EB 2D 3E 00 32 7A 2D 32 E3 2D D2 A3 00 2A 39 30
0E50 7E FE 01 CA 80 00 B7 CA 68 00 CD 9B 28 EB 21 3B
0E60 30 CD CF 1B EB DA 50 00 2A 39 30 3A 2C 30 B7 CA
0E70 75 00 2A 2F 30 36 01 CD 06 05 01 1C 03 C3 F1 01
0E80 22 37 30 3A 2C 30 B7 CA 9D 00 2A 2F 30 23 CD B5
>

```

```

DH 13A0,60
13A0 E5 32 2C 30 3E 01 32 26 30 CD D2 2A CD 8D 1B 3E
13B0 01 CD 1C 00 DA A0 01 4F CD B5 18 D5 23 23 7E E6
13C0 7F FE 02 C2 B8 01 7E E6 80 B1 4F 23 7E B7 CA D8
13D0 01 E1 D1 D5 E5 F5 2A 3B 30 CD 29 09 F1 BC DA E2
13E0 05 7C 0C 0D F2 EA 05 C6 01 1F E1 D1 06 01 CD 22
13F0 00 DA D8 01 37 C3 33 00 CD 8D 1B E5 E5 3E 01 CD
>

```

Listing 2: Dump of -BASIC file

your origins in the offset selection number array O which is the heart of this program. O is first initialized (in line 1290) to -256 in each position of the array. This guarantees that any illegal differences are detected by trying to write a negative byte value to the minus file (an OUT OF BOUNDS ERROR in line 2190).

The differences calculated from each byte in both files cause the stored offsets to be added to the byte from the minus file (line 2190). For example: A difference of 14 (0EH) will add the offset in O(14), a difference of 232 (E8H) will add O(232), and a difference of 1 (01H) will add O(1). O(14) will contain BASIC's new origin offset, O(232) will contain the boot PROM's new origin offset, and O(1) will contain the DOS's new origin offset.

A further complication is the need to handle negative differences and offsets. This will allow relocations to smaller addresses. Obviously, a difference of -14 will cause an OUT OF BOUNDS ERROR when we try to access O(-14) which cannot exist. The O array values are also offset with an additional bias of 45 (in variable K). 45 is 2DH which is the largest one needed by any system software (you cannot have a negative difference of more than 2DH for BASIC and the utilities because it will not be negative any more! The sign changes when the address changes from 0H to FFFFH).

Note that the 2.1.1 version of MOVER is almost identical to 5.2's version. Two additional offsets have been added: for BASIC and the utilities at 1000H, and at 2600H for hard disk software. Use either version of MOVER, but remember that each version has a preferred origin for BASIC and the utilities! Our relocation of SECRETARY can use either version because the 0EH and 2DH offsets are supported.

Relocating SECRETARY

Now we can relocate SECRETARY! HUH?!?! Oh, I did leave out where the minus SECRETARY file comes from, didn't I! No, I did not change all of those bytes in the 2D00H version by hand until I was blue in the face! My computer is quite powerful even if it is not a 16-bitter! I wrote a simple program to do the hard work for me.

INSUA mailed us a copy of the 2D00H version of SECRETARY, but that is not

enough to create a minus file. In my NOTES TO A SECRETARY (DISKETTE) in Compass, Vol. IV, No. 3, p. 8 I offered to anyone who wants it, a copy of the E00H version of the same, identical program. AH HAH!! By comparing each byte in the two versions, we can create the minus file! This is what Program ASSYZERO does. The original test version contained only 23 lines (and six of those are REMs or not needed for a run). The listing of ASSYZERO included with this article is more friendly, but it does run slower than the test version (at least you do not have to edit the program to change the file names or offsets!). ASSYZERO is actually a simpler program than MOVER.

Using ASSYZERO

Note the comments at the beginning of the program! Load it and run it with any identical, object code programs that were assembled for different load addresses. The first file name is the program whose bytes will end up in the output file. I made a copy of the E00H SECRETARY under the file name SEC0E. Likewise, the second version's file name was SEC2D. The output file name can be anything you like, but would normally be -something. I used -SEC0E because the first version's file name was SEC0E. The output file will be of TYPE 3.

Next we enter the full four-digit hex load addresses for the two versions in the same order as the file names. The low bytes must be 00H which is the normal case (loads on a page boundary). The program will calculate the offset between the two versions' high bytes and the decimal load address for each version.

The heart of ASSYZERO is found in lines 880-1030:

880...We will READ and WRITE every byte in the first version's file

910...READ one byte from both files

940...Is the difference equal to that between the two version's high bytes (2DH-0EH=1FH or 31 decimal). The sign is ignored. If not relocatable, this byte is then checked for being a boot PROM byte in line 1030

990...Relocate this byte. If it is within the SECRETARY program, we subtract the

first file's offset of 0EH (14 decimal) which gives the same high byte as if we had assembled the output file for 0H (as per Point 2 above). If this byte is below the program, it must be within the DOS and will be zeroed (as per Point 3 above). Note how the high byte addresses within SECRETARY are handled. If the byte is 0EH, it will be zeroed; if 0FH, it will become an 01H; if 10H, it will become an 02H; etc. This exactly matches Point 2 above

1000..A relocated byte cannot be in the boot PROM, so just write it (in line 1270) and do the next byte (line 1280)

1030..Non-relocatable bytes are checked for a boot PROM high byte address (E8H is 232 decimal). If not, just WRITE it out and do the next byte. Lines 1070-1240 print the actual hex address where this byte was found. I do this to tell you about possible boot PROM high byte addresses.

ASSYZERO will not relocate these (contrary to Point 3), but by telling you about it you can use the Monitor to zero the real boot PROM addresses. Most of you have the standard E800H PROM and do not need to relocate these bytes!

SECRETARY will print the following seven addresses: 10F1H, 12BDH, 15E3H, 18B3H, 2CB5H, 2CE3H, and 37D3H. All are low address bytes except the one at 12BDH which is a high byte. This follows the word BOOT in the command table and tells SECRETARY to jump to E800H (the low byte comes first) when you command a (re)boot. If you want MOVER to relocate this byte to anything other than E8H, you will have to load the minus file at the first version's load address (E00H), run a Monitor that will not overwrite the file in memory, change this byte with a DS 12BD (prints E8=, enter 0 then a RETURN), and save the modified file. Remember, most of you do not need to bother with this!

Miscellaneous Warnings!

Please excuse my repeating here, but I want to make a number of warnings clear about the use of ASSYZERO:

1. Always use working copies of the files, never your only or master copy!

2. You must have two versions of the same, identical program which load at different addresses! If the programs are not identical, MOVER will calculate a bad difference and try to write a negative byte to the minus file giving an OUT OF BOUNDS ERROR. If you try to relocate a copy which has been run or configured, you may cause this problem. Always use a copy of the master diskette's program. Note that some storage location bytes will not affect the relocation, but MOVER may object in the same manner. I leave you to fix this problem on your own!

Relocating SECRETARY with MOVER

When you have created the minus file version of SECRETARY you may need to copy it to a minus file of the same name as the plus file before you run MOVER (the file names must match except for the minus sign). You probably want to keep a copy of the minus file for use later on at different load addresses.

If you relocate SECRETARY for use on the Advantage Computer, the DOS's origin is at E000H, but you should give this origin as F800H which is where this DOS's jump table is located. SECRETARY's origin is 0H.

The Advantage Version Must be Modified!

If you relocate a version of SECRETARY for the Advantage Computer, you will not be able to do anything with it! SECRETARY will not access double-density files until it is CONFIGured! The Advantage cannot access single-density files.

Another problem is that CONFIGure will write a zero to every byte above SECRETARY looking for the boot PROM which was phantomed out and replaced with memory for the DOS! The DOS will be zeroed too! Four bytes have to be changed to prevent CONFIGure from destroying the DOS.

Load a running copy of SECRETARY at 0H and make the following changes with the MC000 Monitor (DS 2F37, etc.):

1. Change the byte at 2F37H from 33H to DFH
2. Change the three bytes at 2F38-2F3AH from AF 77 7E to C3 49 2F

Change 1 will change the starting address for searching for the end of memory from the first byte ABOVE SECRETARY (33FFH) to the last usable byte BELOW the DOS (DFFFH). The storage location for high memory is at 37H and an FF DF will be stored there when you CONFIGure this version (I thought that this was a cute modification!). Change 2 will bypass the code which looks for the top of memory and jump to the instruction which saves the high memory address. Save this modified version and then run it. CONFIGure it and note how fast the high memory address is found (no wait at all)! Continue with the other questions and you will have a version which runs on the Advantage! Remember that you can only save the CONFIGured version to a file named SECRARY!

One minor problem is that you cannot type the command BOOT because the boot PROM does not exist except on start up (it will jump into the DOS at the wrong address and blow the DOS up). I also recommend that you change the two bytes in the command table at 4BCH-4BDH (just after the word BOOT) from 00 E8 to 28 F8. This will cause a jump to the DOS' warm entry point instead of the PROM. Another minor problem is that SECRETARY and the Advantage DOS both use # for the input prompt! It is easy to get confused about which program you are running. You can change the 23H bytes (#) in BBH and 304H to whatever you prefer. A major problem is that the Advantage system software is

not relocatable and did not include a copy of MOVER (see below)!

Another Offer!

Again I make my offer to supply any of the files mentioned in this article including the E00H version of SECRETARY which INSUA has not said a word about. Send me a diskette and \$3.00 for postage and handling (or an extra \$5.00 for one of my diskettes). I will include all of the files plus the corrected files from my NOTES TO A SECRETARY (DISKETTE).

Note that I do not guarantee the relocated version of SECRETARY to work with Release 4 or earlier DOSes, although you are welcome to try it! Also note that if you have destroyed your only copies of the system software minus files, ASSYZERO can recreate them for you!

Copyright Notice and Distribution Rights

You may remove all REMarks and unnecessary spaces (which take up over 45% of this program), but do not remove my copyright notice in Line 40. This program may not be sold in any way or in any form by anyone (except INSUA) including user group diskettes which have any fee connected with them. I do authorize free distribution of this program ONLY IN ITS COMPLETE FORM to anyone you wish.

#

```

10 REM PROGRAM ASSYZERO
20 REM WRITTEN BY SAUL G. LEVY, TUCSON, ARIZONA, JANUARY 5-7, 1985
30 REM LAST CHANGED JANUARY 7, 1985
40 REM COPYRIGHT (C) SAUL G. LEVY 1985
50 REM
60 REM
70 REM DIMENSION
80 DIM H$(16)
90 REM
100 REM DEFINE FIXED VALUES IN PROGRAM
110 P=232 \ REM BOOT PROM HIGH BYTE (E8H)
120 H$="123456789ABCDEF0" \ REM STRING OF HEX DIGITS PLUS THE 0 DIGIT
130 REM
140 REM PROGRAM TITLE, CLEAR SCREEN FOR ADM-3A
150 ! CHR$(26),"PROGRAM ASSYZERO"
160 !
170 ! "This program will allow you to relocate ANY object code program on the"
180 ! "condition that you have TWO COPIES of it ASSEMBLED for DIFFERENT load"
190 ! "addresses! The two versions are compared to allow the relocation"
200 !

```

```

210 ! "Program ASSYZERO will reorigin the FIRST file to the OUTPUT file. The"
220 ! "OUTPUT version is equivalent to having assembled it for 0000H PLUS all"
230 ! "HIGH BYTE REFERENCES to the DOS are also changed to a 0 byte. The new"
240 ! "version CANNOT be run in any way!"
250 !
260 ! "Normally, the BOOT PROM HIGH BYTE (E8H) would also be changed to a 0"
270 ! "byte, but ASSYZERO will just LIST the ADDRESSES of E8H bytes so you"
280 ! "can check them manually (this should not be needed)."
```

```

290 !
300 ! "The actual relocation is performed by North Star's MOVER program which"
310 ! "was included with Release 5.2 and 2.1.1 system software. Either"
320 ! "version can be used"
330 !
340 !
350 REM
360 REM ENTER 3 FILE NAMES, CHECK THAT 1ST 2 EXIST, CREATE 3RD IF NEEDED, OPEN
370 REM FILES
380 INPUT " FIRST VERSION'S FILE NAME: ",F1$
390 IF FILE(F1$)=-1 THEN 380
400 OPEN #1%1,F1$,S
410 INPUT "SECOND VERSION'S FILE NAME: ",F2$
420 IF FILE(F2$)=-1 THEN 410
430 OPEN #2%1,F2$
440 INPUT "          OUTPUT FILE NAME: ",F3$
450 IF FILE(F3$)=-1 THEN CREATE F3$,S
460 OPEN #3,F3$
470 REM
480 REM ENTER LOAD ADDRESSES FOR THE 2 VERSIONS
490 !
500 !
510 ! "ENTER LOAD ADDRESSES FOR FIRST AND SECOND FILES (FOUR DIGITS EACH)."
```

```

520 ! "THE LAST TWO DIGITS MUST BE 00!"
530 !
540 INPUT "LOAD ADDRESS FOR FIRST FILE: ",H1$
550 IF LEN(H1$)<>4 THEN 540
560 IF H1$(3,4)<>"00" THEN 540
570 INPUT "LOAD ADDRESS FOR SECOND FILE: ",H2$
580 IF LEN(H2$)<>4 THEN 570
590 IF H2$(3,4)<>"00" THEN 570
600 REM
610 REM CONVERT H1$ ADDRESS TO DECIMAL LOAD ADDRESS L AND THE HIGH BYTE OF H1$
620 REM TO DECIMAL OFFSET F1 FOR 0000H ORIGIN
630 L=0 \ F1=0
640 FOR I= 1 TO 2 \ REM NUMBER OF HIGH BYTE DIGITS IN H1$
650 FOR J= 1 TO 15 \ REM NUMBER OF DIGITS IN H$ (IGNORE THE 0)
660 IF H1$(I,I)=H$(J,J) THEN L=L+J*16^(4-I)
670 IF H1$(I,I)=H$(J,J) THEN F1=F1+J*16^(2-I)
680 NEXT J
690 NEXT I
700 REM
710 REM CONVERT THE HIGH BYTE OF H2$ TO DECIMAL VALUE F2
720 F2=0
730 FOR I= 1 TO 2 \ REM NUMBER OF HIGH BYTE DIGITS IN H2$
740 FOR J= 1 TO 15 \ REM NUMBER OF DIGITS IN H$ (IGNORE THE 0)
750 IF H2$(I,I)=H$(J,J) THEN F2=F2+J*16^(2-I)
760 NEXT J
770 NEXT I
780 REM
```

```

790 REM CALCULATE DIFFERENCE BETWEEN THE TWO HIGH BYTES
800 D=ABS(F1-F2)
810 REM
820 REM MAIN LOOP TO READ FILES
830 !
840 !
850 ! "RELOCATING!"
860 !
870 ! "POSSIBLE BOOT PROM BYTE AT:"
880 FOR I= 1 TO S*256 \ REM NUMBER OF BYTES IN THE 1ST FILE
890 REM
900 REM READ ONE BYTE FROM BOTH INPUT FILES
910 READ #1,&B1 \ READ #2,&B2
920 REM
930 REM NOT RELOCATABLE? THE DIFFERENCE MUST MATCH THE CALCULATED VALUE
940 IF ABS(B1-B2)<>D THEN 1030
950 REM
960 REM IT IS RELOCATABLE!
970 REM IF A PROGRAM HIGH BYTE, THEN RELOCATE IT BY SUBTRACTING THE OFFSET.
980 REM IF A DOS HIGH BYTE, THEN ZERO IT
990 IF B1>=F1 THEN B1=ABS(B1-F1) ELSE B1=0
1000 GOTO 1270
1010 REM
1020 REM BOOT PROM HIGH BYTE?
1030 IF B1<>B2 OR B1<>P THEN 1270
1040 REM
1050 REM IT IS A BOOT PROM HIGH BYTE! FORM ITS DECIMAL ADDRESS BY ADDING THE
1060 REM DECIMAL OFFSET VALUE TO THE BYTE POINTER I-1 FOR BOTH OFFSETS
1070 A1=I-1+F1*16^2 \ A2=I-1+F2*16^2
1080 REM
1090 REM CONVERT DECIMAL ADDRESSES A1 AND A2 TO HEX STRINGS
1100 FOR J= 1 TO 4 \ REM NUMBER OF HEX ORDERS OF MAGNITUDE
1110 H1(J)=A1/16^(4-J) \ H2(J)=A2/16^(4-J) \ REM DECIMAL VALUE OF 1 HEX DIGIT
1120 A1=A1-INT(H1(J))*16^(4-J) \ A2=A2-INT(H2(J))*16^(4-J) \ REM REMAINDER
1130 H1(J)=INT(H1(J)) \ H2(J)=INT(H2(J)) \ REM REMOVE FRACTION
1140 IF H1(J)=0 THEN H1(J)=16 \ IF H2(J)=0 THEN H2(J)=16 \ REM HEX 0 IS SPECIAL
1150 NEXT J
1160 REM
1170 REM FORM HEX STRINGS
1180 B1$="" \ B2$="" \ REM NULL BOOT PROM ADDRESS STRINGS
1190 FOR J= 1 TO 4 \ REM NUMBER OF DECIMAL DIGITS TO BE CONVERTED
1200 B1$=B1$+H$(H1(J),H1(J)) \ B2$=B2$+H$(H2(J),H2(J)) \ REM FORM 1 HEX DIGIT
1210 NEXT J
1220 REM
1230 REM PRINT BOOT PROM ADDRESS STRINGS
1240 !#0,B1$,"H IN ",H1$,"H VERSION, ",B2$,"H IN ",H2$,"H VERSION"
1250 REM
1260 REM WRITE ONE BYTE TO OUTPUT FILE
1270 WRITE #3,&B1,NOENDMARK
1280 NEXT I
1290 REM
1300 REM FINISHED
1310 !
1320 !
1330 ! "FINISHED"
1340 CLOSE #1 \ CLOSE #2 \ CLOSE #3
1350 END
1360 REM LAST LINE OF ASSYZERO

```

Banking on your Computer

By Alan H. Nelson

Banking by personal computer? It's here, at least in the San Francisco Bay Area. For about a year now, Bank of America has provided HomeBanking services which can be purchased by anyone with a local bank account, a personal computer, and eight spare bucks a month.

The only special equipment needed is a modem, and of course a modem program. MODEM712 (for CP/M), or TELESTAR (for DOS), both available through the INSUA library, are fine. BofA accepts calls from both 300 and 1200 baud modems, so whatever you now have will probably work. Use terminal mode, and save a session on disk (Ctrl-Y in MODEM7) if you want to keep a complete record of your transactions.

How does it work?

1) You hook up your modem, then dial a local number given you by the bank. At a prompt, you type in the name of the service you require, in my case HOMEBANKSF. (All typing can be done in either capitals or lower-case letters.) Then another prompt asks for your account number. In this case, it is the same as my Versateller number--in fact, since I didn't have a cash-withdrawal card previously, I had to sign up for one in order to use HomeBanking. Finally, you are asked for a password, identical to a PIN code, but unfortunately or fortunately, not the same as my cash-withdrawal PIN code. (Another number to memorize!)

2) After you have succeeded in gaining access to the system by following this procedure (which is more or less routine for time-sharing services), a prompt appears. (With 300 baud it appears immediately; with 1200 baud it is necessary first to hit the letter "a", which the receiving computer deciphers as a coherent 1200-baud signal, setting its own baud rate accordingly.)

3) From here on out, it's mostly a matter of carrying on a dialogue with the host computer. A sample session follows, with

numbers changed to protect the innocent (that's me!) User-typed material is shown in bold face.

A Typical Session

(DIAL UP)

-1234-567-
please log in: **homebanksf**

host: call connected
ENTER ID:)**123456789**
PASSCODE: (doesn't show)
WELCOME TO HOMEBANKING
COPYRIGHT 1983 BANK OF AMERICA NT&SA
CHECK ANNOUNCEMENTS FOR INFORMATION
ON NEW HOMEBANKING SERVICE FEATURES
AND FOR IMPORTANT NEWS ABOUT IRAS.

**** MAIN MENU ****

0 LOGOFF
1 PAY BILLS
2 TRANSFER FUNDS
3 BALANCE INQUIRY
4 REVIEW OR CANCEL
5 ELECTRONIC STATEMENT
6 ELECTRONIC MAIL
7 ACCOUNT MAINTENANCE
? COMMAND (CMD) LIST
ENTER: 3

****BALANCE INQUIRY****

1 CHECKING
2 SAVINGS
3 VISA
4 MASTERCARD
SELECT ONE: 1

CHECKING BALANCES
ENTER SECTION NBR OR P FOR NEXT PAGE
ACCOUNT 12345-67890
\$1,234.56
ENTER: 2

SAVINGS BALANCES
ENTER SECTION NBR OR P FOR NEXT PAGE
ACCOUNT 12345-67890
\$1,234.56
ENTER: 3

VISA BALANCES

ENTER SECTION NBR OR P FOR NEXT PAGE
ACCOUNT 1234-5678-9012-3456
AMT DUE \$1,234.56 CR AVL \$789
ENTER: q
** MAIN MENU **
0 LOGOFF
1 PAY BILLS
2 TRANSFER FUNDS
3 BALANCE INQUIRY
4 REVIEW OR CANCEL
5 ELECTRONIC STATEMENT
6 ELECTRONIC MAIL
7 ACCOUNT MAINTENANCE
? COMMAND (CMD) LIST
ENTER: 1

PAY FROM 12345-67890
SELECT A PAYEE OR
P FOR NEXT PAGE
1 BA VISA
1234-5678-9012-3456
2 PACIFIC BELL
1234567890123
3 SEARS
1234567890123
4 P G AND E
ABC12345678
5 P G AND E
DEF12345678
6 SHELL
123456789
SELECT ONE: 4

ENTER T FOR TODAY'S BUSINESS DATE
OR MM/DD/YY FOR FUTURE DATE
DATE: t

AMOUNT\$: 123.45

FROM 12345-67890
PAY P G AND E
ABC12345678
ON 01/01/85 \$123.45
PLEASE CONFIRM
Y OR N: y

PAYMENT ACCEPTED
TRANSACTION NUMBER 123456
ANOTHER PAYMENT?
Y/N OR CMD: y

PAY FROM 12345-67890
SELECT A PAYEE OR
P FOR NEXT PAGE
1 BA VISA
1234-5678-9012-3456
2 PACIFIC BELL
1234567890123

3 SEARS
1234567890123
4 P G AND E
ABC12345678
5 P G AND E
DEF12345678
6 SHELL
123456789
SELECT ONE: 5

ENTER T FOR TODAY'S BUSINESS DATE
OR MM/DD/YY FOR FUTURE DATE
DATE: t

AMOUNT\$: 12.34

FROM 12345-67890
PAY P G AND E
DEF12345678
ON 01/01/85 \$12.34
PLEASE CONFIRM
Y OR N: y

PAYMENT ACCEPTED
TRANSACTION NUMBER 123457
ANOTHER PAYMENT?
Y/N OR CMD: n

** MAIN MENU **
0 LOGOFF
1 PAY BILLS
2 TRANSFER FUNDS
3 BALANCE INQUIRY
4 REVIEW OR CANCEL
5 ELECTRONIC STATEMENT
6 ELECTRONIC MAIL
7 ACCOUNT MAINTENANCE
? COMMAND (CMD) LIST
ENTER: 6

0 READ ANNOUNCEMENTS
1 READ MAIL
2 SEND MAIL TO THE VERSATEL CENTER
ENTER: 1

YOUR MAIL BOX IS EMPTY,
ENTER 0 TO READ ANNOUNCEMENTS
ENTER 2 TO SEND MAIL OR
ENTER =MM FOR MAIN MENU

ENTER: 2

FROM: ALAN H NELSON
TO: VERSATEL CENTER
SUBJECT: emporium sellwell

....5...10...15...20...25...30...35

1 What's wrong with Emporium
2 Sellwell? I have been trying since
3 December to link my account so
4 I can pay them by computer.
5 Result? NOTHING!!!
6 Alan Nelson
7

1 SEND THIS MESSAGE
2 REVIEW THIS MESSAGE
3 CANCEL THIS MESSAGE
SELECT ONE: 1

YOUR MAIL MESSAGE WAS SENT ON
02/03/85 AT 12:23. WOULD YOU LIKE
TO READ OR SEND MAIL?
Y/N OR CMD: n

** MAIN MENU **
0 LOGOFF
1 PAY BILLS
2 TRANSFER FUNDS
3 BALANCE INQUIRY
4 REVIEW OR CANCEL
5 ELECTRONIC STATEMENT
6 ELECTRONIC MAIL
7 ACCOUNT MAINTENANCE
? COMMAND (CMD) LIST
ENTER: 0

YOUR HOMEBANKING SESSION
HAS BEEN CONCLUDED.

host: call cleared (c 0,d 0):
dte originated

How safe it it?

Theoretically home computer banking is quite safe, since there is no cash withdrawal, only monetary transfers from one linked account to another. Supposing someone discovered your code--he could probably make a mess of things by transferring money around at random and paying more to an authorized recipient than you really owed. Nevertheless, unless the intruder maintained an account approved for linking, there is no apparent way he could personally benefit. Certainly no private individual would be granted merchant account status; Joe can't transfer funds to Harry with this system, though he could transfer money to Sears.

Is it worth the cost?

As I currently use the service, it is probably not worth eight dollars a month. After all, even with stamps costing 22 cents, the total postage I spend on the accounts would be about two dollars, leaving a negative balance of six dollars. If I did keep a closer eye on my various funds, however, it probably would pay for itself easily.

Checking accounts nowadays often require a minimum average balance to avoid per-check fees. The best way to play this game with your bank is to keep just this average and no more; any excess amount should be kept in an interest-drawing savings account. By phoning in once or twice a week it is possible to move money from checking to savings or vice versa to keep as much as possible in savings.

Another nice and potentially money-saving feature is the plan-ahead payment on installment accounts. VISA is an example of this. My VISA account payment is always due about the 3rd of the month. My salary check comes in the first, or sometimes the last day of the previous month. I can instruct the account to transfer the funds from checking to VISA on the first of the month; that way the payment should always be covered, and I can get the maximum extension on my VISA account without having to pay interest.

HomeBanking is similarly convenient for mortgage payments or any other routine transfers of funds. I don't know how far ahead it is possible to plan such payments, but clearly it is possible to instruct the program months in advance of the actual payment.

Another more obvious convenience is that it is possible to check the balances in your account at the same time you are balancing your check book, which (if you're like me) may well be at night, when the bank is closed.

More automation?

HomeBanking maintains a pretty complete record of payments and deposits from your account. Can it be tied into an automatic financial planning program? I haven't tried anything in this line yet, but

I am pretty certain it would work. I can imagine a dBASE program which will accept the results of a session saved into a .TXT file, scan the lines for tell-tale alphabetical or numeric strings, store this information into a more organized file, check for duplications (in check numbers,

etc.), and give current balances etc.

I haven't written such a program, but it shouldn't be too difficult. I wonder whether anyone out there has done something like this?

#

Vendors

We would like to announce our prototyping and extender boards for the North Star Advantage. Each board is the same size as the Advantage Serial I/O card, and is industrial quality, with gold-plated fingers, plated through holes, and pre-tinned.

The Extender includes a female edge connector on the top; it is used to raise a board above the bus for easy access during debugging and testing. It also contains two small prototyping areas (one 1/2 inch by 5 inches, the other 1/4 inch by 5 inches).

The Prototyping board has heavy power and ground traces around the edges of the board, with holes for tantalum capacitors near the power feed-in. A DB-25 or DB-37 may easily be mounted, as signal and mounting holes are provided. The Prototyping board may be used to develop your own custom circuits: a real-time clock, colour graphics, RAM disk, A/D converter, etc.

Doug Anderson Techno-Products Inc.
947 Richmond Road
Ottawa, Ontario
Canada K2B 6R1
(613) 722-0690

\$ \$ \$

North Star Computers Announces:

8087 Arithmetic Processor Now Available for the North Star DIMENSION. The North Star DIMENSION can now outperform a standard IBM PC by as much as 500 percent in many CAD/CAM applications, as a result of utilizing the Intel 8087 Numeric Data Co-Processor chip now available for the DIMENSION from North Star Computers, Inc. The 8087-2 chip in the North Star DIMENSION

now makes it practical for each engineer in a company to have his own microcomputer-based CAD/CAM application at his desk.

The high-performance 8087-2 chip is an add-in option for the existing multi-user, IBM PC/XT compatible DIMENSION line. Each of up to 12 workstations on the DIMENSION features an 8MHz 8088-2 processor, and can now also have an 8MHz 8087-2 co-processor for floating point arithmetic.

Previously, IBM PC/XT compatible CAD/CAM applications had to be used on single-user PCs, which posed cost and performance problems for engineering departments and small firms. With the DIMENSION, every user can now have his or her own terminal at a cost of approximately \$2,500 per user. The inherent benefits of multi-user systems allow these users to work on the same projects concurrently.

The new DIMENSION Arithmetic Processor Module consists of an 8087-2 integrated circuit and the processor board to hold it for a suggested list price of \$550. It functions with the current release of the DIMENSION operating system and requires no software modifications.

In a test of the popular AutoCAD-86 software by Autodesk, Inc., North Star found that typical performance of a DIMENSION workstation with the new 8087-2 module ranged from 350 percent to 550 percent better than that of a standard IBM PC. For example, it took 8.0 seconds for the DIMENSION to complete a typical AutoCAD function that required 35.0 seconds on the IBM PC.

The North Star DIMENSION has been shipping since May and is sold through more than 350 authorized DIMENSION resellers throughout the world.

North Star Computers
14440 Catalina St.
San Leandro, CA 94557
(415) 357-8500

GRAPHICS

SAS TYPEWRITER GRAPHICS FOR PROGRAM EVALUATION

Warren Lambert, PhD

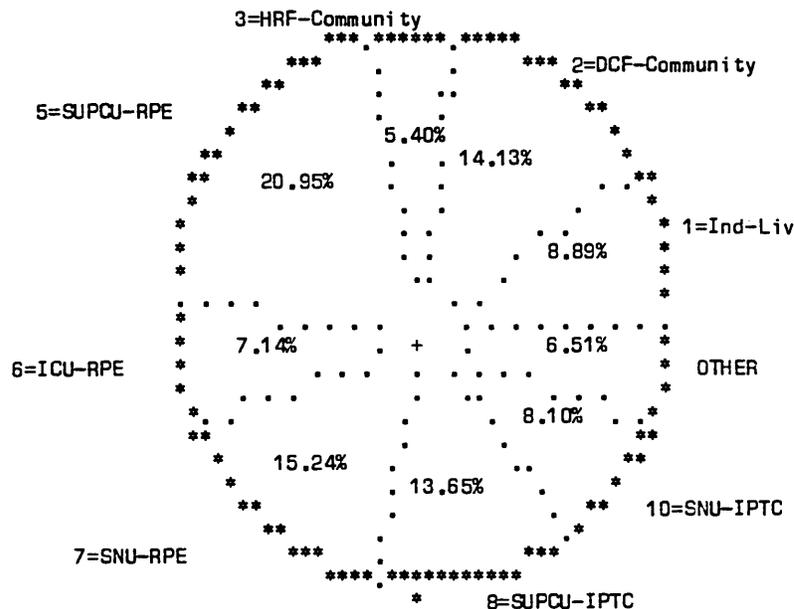
"FACTS ABOUT PATIENTS" is a crucial theme in mental health program evaluation, but state mental health systems have vast populations of patients. Organizing such massive amounts of information requires mechanical help. Excellent statistical packages of "canned" programs, such as SPSS (Statistical Package for the Social Sciences) or SAS (Statistical Analysis System) enable the computer user to function quite well with occasional help from the professional programmer.

While SPSS statistical programs are quite useful, no program evaluator of sound mind would simply send raw output from subprogram DISCRIMINANT, RELIABILITY, or even CROSSTABS to high level administrators, who often distrust complex statistical procedures. SPSS and SAS both have new options for color graphics, but it will take time for the expensive color terminals to be widely available, if they ever do. SAS, however, has a procedure called "PROC CHART" which can be used to prepare printer or typewriter charts right now, and the computer user with little training in programming may find PROC CHART friendly and accessible after a little practice. Saved SPSS data files can be analyzed by SAS quite easily.

PROC CHART can produce printer graphics in five styles:

*** P I E charts are the simplest charts, and not very flexible. However, the nonstatistical data consumer can read them easily. Figure 1 shows distribution of ten levels of care indicated by a Care10 survey in a Tennessee hospital.

FIGURE 1.
Percentage pie chart of Levels of care needed by 692 inpatients at a psychiatric hospital in Tennessee



COMPASS INDEX

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
Covilli	D. XMODEM Transfer Protocol	4.1	2		Description of standard MODEM7 protocol
Maguire	J. The Phantom Signal	4.1	5		History and applications of phantom line--pin 67
Nelson	A. Dummy Lines	4.1	7		Technique for forcing the printing of a "blank" line
Porter	R. Commentary	4.1	8		Dumping BASIC variables and installing quad drives
Hogan	S. Computer Pollution Solution	4.1	9		Fitting a fan filter to the Horizon
Lambert	W. Alien Equipment	4.1	13		Ted Carnevale reviews alien S-100 boards
Lambert	W. WordStar Block Letters	4.1	15	Y	Creating large block letters w/ WS and MailMerge
Maguire	J. Beyond 64K	4.1	16		Puget Sound Computer Service creates a portable
Maguire	J. 3 Laws of Bank Switching	4.1	17		Principles of bank-switching
Morgenstern	L. MEMTODAT	4.1	19	Y	Clip Tip
Appleby	D. Vendors' Column	4.1	21		BASIC program to create DATA statements
Nelson	A. Appleby on TurboDOS	4.1	23		Printer Buffer, N*CAT, AUTOTYPE editor
Swenson	L. Advantage Graphics	4.1	24		First impressions of TurboDOS
	A. Ribbonizer Revisited	4.1	25		List of vendors
	L. Quic-n-easy	4.1	26		Followup of earlier review (3.1 pp 29-30)
Levy	S. User's Visit to North *	4.1	27		Short review of program to assist in writing programs
Noll	S. Commentary	4.1	28		1983 visit--impressions and opinions
	R. More on APC BASIC	4.1	29		INSUA policy, APCBASIC, and Advantage graphics
	A. Whither Micro Design	4.1	30		Readers' reactions to MagicBind and Diablo 630
	J. Correction	4.1	31		Portability of BASICS including N* and APC
	R. Morrow Hard Disk	4.1	32		Problems with a promising disk controller
	B. Installing the Morrow16Mb	4.1	32		Correction to "Two-Column" article (3.4)
	J. Vertical Horizon	4.2	2		Installing the 16Mbyte drive
	J. Disk Emulator: A Review	4.2	5		The Morrow 16 Mb hard disk drive
	J. File Length in DOS	4.2	6	Y	Mounting the Horizon vertically for convenience
	J. The Blank PC Board Option	4.2	7		The IS-100 Disk Simulator from Digital Research
	A. Pizza Calculator	4.2	8	Y	A Solution to some End-Of-File Problems
Ahlgren	R. Modification of 32K Board	4.2	9		Discussion of kits offered with boards but no parts
Shulkin	R. Softwarecontrolled Switch	4.2	12		Program to determine price of pizza by the square inch
Maguire	J. Relocating Disk Control'r	4.2	14	Y	Increasing available memory through board fixes
Ingram	D. File Conversion Program	4.2	18	Y	Switching relays controlled from unused port
Shearer	J. N* BASIC Pre-processor	4.2	26	Y	Software control of the disk controller location
Levy	S. NSPRE: A Review	4.2	30	Y	Convert ASCII and BASIC files from DOS to CP/M
Shearer	J. NSPRE: A Review	4.2	36		Write BASIC programs without linenumbers!
Maguire	J. Letters to Joe	4.2	37		Review of Jim Shearer's Pre-processor
Bryant	J. Bubble Disk on the Horizon	4.3	2		Jim Shearer's response to Saul Levy's review
Allen	D. Graphics from TurboPascal	4.3	5	Y	Advice for all letter-writers
Staponski	R. Low Cost Word Processor	4.3	6		Installing an S-100 Bubble Memory Board
Levy	S. Notes to a SECRETARY	4.3	8		A Turbo Pascal Pascal program for Advantage graphics
Maguire	J. S-100 Extended Addressing	4.3	21		Review of and Introduction to SECRETARY
Lambert	W. Alien Equipment	4.3	24		Supplement to the SECRETARY manual
Noll	S. Commentary	4.3	26		Upgrading the Horizon for 64K-plus memory
	Vendors Column	4.3	29		Randall Staponski reviews Epson buffer boards
	Letters to the Editor	4.3	32		Computer Insurance, SCAN & REN, Graphics Fischer Computer Systems, Angwin, CA

COMPASS INDEX

AUTHOR	TITLE	VOL	PG	LST	CONTENTS
Beaver	B. dBASE a la Menu	4.4	2	Y	A general menu for any dBASE master program
Maguire	J. Run your Horizon at 6Mhz	4.4	4		Upgrade your Z-80 chip to speed up your system
Saunders	W. Commentary	4.4	6		The Phantom Signal and Fan Filters for the Horizon
Stuhlman	D. Horizons and Apples	4.4	7		Bench-test comparisons for computers and programs
	Bugs	4.4	12		Hugh Power reports a bug in 5.2 DOS
Nelson	A. Look before you Leap	4.4	13		PC-DOS compared to CP/M
	Advantage Answers	4.4	15		Function keys, windows, 8/16, Documentation
Hough	T. Commentary	4.4	16		HDOS File Manager and North Star BASIC
Maguire	J. Disk Controller PROMS	4.4	17		Decoding North Star's Disk Controller PROMS
Levy	S. File-Size Changing	4.4	19	Y	FILESIZE Program
Prohaska	C. Disk Simulator Patches	4.4	28	Y	Patches offered for the LS-100 Disk Simulator
	Vendors Column	4.4	29		Micro Support, TIS-APL, Eye-Care CR-39, TurboDOS
	Letters to the Editor	4.4	31		
Coudal	E. The Trailing Edge	4.5	2		Reminiscences: 6 Years and a N* User
Bryant	J. 38,400 Baud: Hardware	4.5	3		Higher Baud rates for your terminal
Bryant	J. 38,400 Baud: Software	4.5	5		Higher baud rates for your terminal
Nelson	A. BIBLIOGRAPHY: A Review	4.5	8		Program for databases with variable-length fields
Dounson	P. Commentary	4.5	12		More on SECRETARY
Nelson	A. Sound Steward: A Review	4.5	13		User's experience installing a fan filter
Lambert	W. Lazy Man's Handshake	4.5	16		Using a "secret" handshake already in DOS
Lind	J. N*SORT with BASIC 5.5	4.5	18		Routines to make N*SORT work with 5.5
Saunders	W. Commentary	4.5	20		On Phantom Signal and Fan Filters
Hough	T. Here's the Meat	4.5	20	Y	Listings left out of last issue
Carnevale	T. Soft-Sector on Horizon	4.5	22		Making the Morrow DJ/DMA work with Horizons
Hightower	C. Computer Books	4.5	24		Reviews of four books on the Z-80
Maguire	J. Dear Joe	4.5	27		Replies to inquiring readers
	Letters to the Editor	4.5	29		
Fischer	R. TurboDOS and Single User	4.6	2		Description of TurboDOS, functions & features
Allen	D. U.S. Robotics Modem	4.6	4	Y	Using & installing USR Modem with MODEM7
Shih	J. Semi-Disk Emulator	4.6	11		Review of RAM disk comparison to floppies
Van Erp	M. Multiple-Line Headers	4.6	13		Multiple-line headers & footers with WordStar
Garst	P. String that Fyle	4.6	14	Y	Using files which are a single long string
Makinson	C. Modification of Baud Rate	4.6	16	Y	A program to modify Advantage baud rate
Levy	S. Relocating SECRETARY	4.6	19	Y	Description of N*'s MOVER, moving SECRETARY
Nelson	A. Banking on Your Computer	4.6	27		Description of BofA's homeBanking
	Vendors Column	4.6	30		

RENEW! RENEW!

RENEW! RENEW! RENEW! RENEW! RENEW! RENEW!

RENEW! RENEW! RENEW! RENEW! RENEW! RENEW!

RENEW!
RENEW!
RENEW!
RENEW!
RENEW!
RENEW!
RENEW!
RENEW!
RENEW!
RENEW!

RENEW! RENEW! RENEW! RENEW! RENEW! RENEW! RENEW! RENEW! RENEW! RENEW!

NOW IS THE TIME TO RENEW YOUR INSUA MEMBERSHIP!

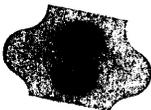
Insua

International NorthStar Users Association

Publishers of The Compass Newsletter

PO Box 2910 • Fairfield, CA 94533

Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203



TATE 276H
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

Postmaster: Address correction requested.
Please enter key number on form 3579.