
SOME THOUGHTS ON BASIC COMPILERSby Burks A. Smith of DATASMITH
Box 8036, Shawnee Mission KS 66208

Elitists in the data processing field, especially educators, tend to make degrading statements about BASIC, claiming it encourages poor programming habits because of its unstructured organization and slow execution. They prefer structured, compiled languages instead with so called "powerful" features that supposedly make programming more efficient. However, the truth is that, when it comes to microcomputers, BASIC is by far the most highly developed language with the most features because it is the de facto standard for micros. Many other languages, while they may have slick structures that teach programming theory quite well, are deficient in the features needed for the production of commercially useful programs, especially for business applications. Most have quite primitive facilities for getting data to the screen or printer, simplistic file access commands, and limited precision.

As far as transportability is concerned, there are really no languages for micros that can be considered to have a "standard" syntax. Even old languages such as COBOL and FORTRAN with well-defined international standards seem to slip into dialects when it comes to implementation on micros because somebody always wants to include "enhancements" that make the language non-standard. Basic is probably the worst when it comes to dialects, and it is my opinion that efforts to standardize it will fail because the standards will be ignored. This may be for the best since hardware technology is changing so quickly that the language must change to take best advantage of the hardware available. Who wants to use a BASIC designed for teletypes and tape storage when you have color terminals and winchester disk systems?

Probably the most transportable programs, at least for computers using the same type of processor (the 8080 compatible family is most common) is machine language. All of these chips speak the same language, and you don't need an interpreter to run the programs. Nobody writes in machine language, so the only two ways to produce a machine language program is with an assembler or a compiler. Assembly language, while producing lightning fast and ultra efficient code, is tedious to use and is best suited for specialized applications that do not require a great deal of changing or maintenance. Compiler languages, on the other hand, allow you to use a high-level language and still produce machine code that, while not as efficient, is many times faster than you would get with an interpreter.

There aren't too many BASIC compilers on the market, but I will discuss the three with which I am familiar. All produce executable 8080 machine code and generally provide significant speed increases of up to ten times what you would expect an interpreted program, provided they are not limited by the speed of peripherals. Obviously, you will see no speed increase in a program that produces mostly printed output, because the speed will be limited by the printer. Compilers provide the best speed increase when executing highly repetitive control structures and data manipulation routines, which are easily constructed in machine language. Floating-point math, trig functions, and other complex math functions are still rather time-consuming. I have made no attempt to compare the speed of compiled programs since, in my opinion, they are all fast enough for almost any application except some types of real-time simulation or other

specialized applications. All require CP/M as an operating system except BASIC-Z, which will run under either CP/M or MDOS.

MICROSOFT BASCOM

Microsoft's BASCOM has been around the longest and has the fewest features. However, it is the only one of the three that is compatible with an interpreted BASIC. This is a very important and useful feature, since compiled programs are not interactive with the programmer. Once the machine language code has been produced, it bears no resemblance to the original source program and can not be traced or interrupted to display the value of variables. This makes development much more difficult, since you can't just change something and type RUN to try again. You must try to guess what the problem might be, change your source program, run the compiler again, and finally see if you fixed the problem. Just correcting one error can take fifteen to thirty minutes.

BASCOM will do anything Microsoft BASIC-80 will do with a few exceptions, all of which degrade the language. First, the compiled Microsoft BASIC doesn't allow a dynamic array dimension. In other words, if you want to DIM an array, you have to do it with a numeric constant as a dimension, not a variable. This gets in the way of many applications, since many times you don't know how big of an array you'll need until the program finds out how much data it has to process. Second, there is no CHAIN statement that allows loading another program segment while retaining current values for variables. This isn't too much of a handicap unless you are running on a system with very limited memory because compiled programs generally take less memory. Third, there is no RESET command. In case you aren't familiar with CP/M, RESET is the command needed to update disk directories in preparation for swapping disks while a program is running. Without RESET, CP/M will terminate your program with a "Read Only" error if you try to update a disk that wasn't in the drive when you started.

Programs compiled with BASCOM require a run-module called BRUN which contains standard support routines for the program. If you want to sell compiled programs, you must also provide the BRUN module and pay Microsoft a royalty for its use. The BASCOM compiler with BRUN lists for \$395.00, but can usually be found for sale at up to 20% off from mail order houses.

SYSTEM-Z'S BASIC-Z

BASIC-Z is probably of the most interesting to Micropolis users because it will run under MDOS and is based on Micropolis Basic syntax. This does not mean that BASIC-Z is totally compatible with Micropolis Basic, however. There are a tremendous number of changes and enhancements in the language, so while it may be possible to test code using Micropolis Basic and then transfer to BASIC-Z, some Micropolis syntax may not work and you will miss some very powerful commands that are not available in Micropolis Basic or any other Basic. Debugging compiled programs is made somewhat easier by the fact that BASIC-Z does provide a debug facility that allows single-stepping and the printing of specified variables at pre-defined points. This feature, combined with a very good program editor that acts like an enhanced version of Micropolis LINEEDIT, makes BASIC-Z easy to use, for a compiler.

One of the major differences between Micropolis Basic and BASIC-Z is in the Input/Output statements. You can no longer OPEN the terminal or printer as a file and use PUT statements for output. Instead, an LPRINTER statement is used to send PRINT statements to the printer, like the Micropolis ASSIGN statement does. This may seem like a step backward, and it does require logical changes in existing programs, but it is offset by enhancements in screen I/O that are extremely powerful. By using an INSTALL utility program, BASIC-Z can be configured for many different

terminals, allowing the language to contain generalized screen handling commands, an editable input statement, a TAB with both row and column as arguments, and even a blinking display mode. This makes BASIC-Z programs very transportable, and allows fancy screens without knowing what terminal you will be using.

The other enhancements are too numerous to mention, but are definitely impressive. In fact, BASIC-Z has more features than any other BASIC I have seen, with the possible exception of IBM System/34 BASIC, which is dedicated to a small mainframe! One feature System-Z likes to advertise is the SORT verb, which will sort an array likely-split. This is indeed unique, but I find it of limited value. It only sorts a one-dimensional array, and the most common use of sorts is the "tagalong" sort whereby a key is used to order a great deal of related data. The SORT verb will only do this to a limited extent, allowing the key to be defined as the left-most X characters of a string of up to 250 characters. To perform a useful tagalong sort, a special array would have to be constructed containing both a key and a pointer. After sorting, the re-ordered pointers could then be used to access data in other arrays or records on disk. I don't know if this would save time or not.

BASIC-Z retails for \$345.00, but our fearless editor, Buzz Rudow, will sell it to MUG members at a discount. Like BASCOM, BASIC-Z requires a run time module (called RUN-Z), but the System-Z folks aren't asking for royalties on it. If you want to sell compiled programs, you can include the RUN-Z and INSTALL programs just by acknowledging the System-Z copyright. You may not, of course, distribute the compiler itself.

DIGITAL RESEARCH CB-80

CB-80 is a new compiler that is CBASIC compatible. As you may know, CBASIC is a semi-compiled language that produces an "intermediate" file that is then interpreted. CBASIC is a good language as far as features are concerned, but it has the disadvantage of having to be compiled coupled with the slow execution speed of interpreted languages. CB-80 is a compiler that takes CBASIC all the way to machine language.

CB-80 is intended for professionals, and is probably the most difficult to use of the three compilers. It has many strengths, however, and is a very high quality product. Source code is entered with a text editor (user supplied) as ordinary text. CB-80 doesn't need a number for every line and unlike CBASIC, it allows alphanumeric labels as jump entry points. As a result, the source code looks more like FORTRAN than BASIC, and very readable code can be produced. The fact that only entry points need have labels or line numbers really makes the structure of the program stand out, and tabs can be used to indent code for definition of logical structures.

Data is stored and retrieved from files by use of a variable list in the manner of Micropolis Basic and BASIC-Z, but files can also be read and written a byte at a time. The ability to treat a disk file as a stream of bytes is similar to the MDOS @RFINX-POS and @WTINXPOS subroutines and is especially well suited for file conversion, transfer, and communications. Other unique features are the ability to determine whether or not a file exists on disk and its size without having to open it first, and a string length of up to 32K. The latter allows buffering of large quantities of data in communications applications.

CB-80's method of producing machine code is the most sophisticated of the three compilers. The CB-80 compiler produces relocatable object code files which are then combined with other relocatable files and libraries of pre-defined functions to produce a machine executable file. A library of standard BASIC functions is provided, but the user can also build his own libraries of routines that, in effect, allow the language to be enhanced to provide any function at all. Assembly

language routines produced with the Digital Research RMAC assembler are fully compatible and can be incorporated directly into the resultant program. The process of producing executable files with a LINK program means that no run-time module is needed because all support routines are part of the main program. An additional efficiency is achieved due to the fact that only those support routines that are actually needed become part of the object program. If you don't use trig functions, for example, the code for them won't be included in your program.

Like BASIC-Z, Digital Research does not require a royalty payment when you sell programs compiled with CB-80. The compiler's price is steep: \$500 will buy the compiler, linker, and library management programs. I haven't seen it discounted yet, either.

DO YOU NEED A COMPILER?

Maybe not. Programs that have a lot of keyboard input, printer output, or are disk-bound will not gain much speed and compiled programs are much more difficult to develop and maintain. Bugs must be fixed by recompiling the whole program, since you usually know how the machine code produced by the compiler works. Even assembly language is easier to patch. On the other hand, if you do more computing than I/O, the speed increase will be spectacular. A file sorting program that I converted from Micropolis Basic to CB-80 ran in minutes instead of hours and I am now able to produce fast utilities in BASIC in a fraction of the time it took me to write them in assembly language.

.....

BUILDING THE CHEAP COMPUTER, PART V

by Zot Trebor

Whew! I'm glad that's over with. Those video things are really crazy. It's nice to turn on the power, latch the disk and have the system come alive. I only wish the routines were in PROM's.

I've described a very cost effective system built around SSM cpu and video boards. Mass storage is taken care of by our faithful Micropolis, leaving only memory and I/O to be defined.

Memory was a snap. After years of fumbling with various types and sizes of memory I finally bought a 64K CMOS RAM board from Digital Research (Computers) of Texas. It solved all of my memory problems. Since I use the upper 8k of RAM for sundry chores other than memory, the board is only populated with 56K. I'm very happy with it and gave Buzz a separate 'product review'. If the interest is there, I'm sure he'll publish it.

The last part of the system is the SSM I/O-4. This is a two-serial port, two parallel port I/O board. Once again, it reflects the Cheap Computer philosophy -- nothing sexy, just plain old-fashioned (in computers?) common sense. The two serial ports are driven by two hardware-strapable USART's; you need to install straps or set switches to program the board for word length, baud rate, number of stop bits, etc. Such chores are handled in software for many boards, which is the reason for the initialization routines in the RES module. I actually prefer the hardware vs software approach for I/O; it's certainly easier to set up, although once set up and running, either type of board gives equally good service. The board presently drives two printers, a Heathkit H-14 (know as The Toad) and a C. Itoh Starwriter II, which is too young to have acquired a nick-name. I'll probably end up calling it The Orphan, considering the lack of support it's had from the dealer and distributor. I'm still trying to get a manual for it, for example. You'd think a nearly \$2,000 printer would come with a manual. Wouldn't you?

Central Processing Unit, Video, Memory, I/O and Mass Storage; that's my Cheap Computer. It handles a variety of applications from word processing to some sophisticated number crunching for light aircraft design; the latter programs are the kind you load before breakfast and check on after lunch to see if it's done yet. After seven years of fooling with micro's I think I've finally found a system that I can count on. The cost of the boards is low enough that I can keep spare boards on hand should a problem occur. Since three of the five boards come from one manufacturer, a trip back to the factory should produce a minimum of finger-pointing. I've only had this configuration up and running for about a year, although some of the boards are much older.. The system averages 6.7 hours of 'on' time per day and my last equipment failure was due to my plugging in a jumper upside down.

The Cheap Computer could be duplicated, in a nice cabinet, for about \$4000 (1982 prices). I'm certainly pleased with it and I hope you enjoyed having it described to you.

Next on the agenda is the software for an old Szerlip PROM programmer. I think the program will work for the SSM programmer as well; perhaps one of you who has an SSM programmer could try it out and tell me.

Also for a future article; the infamous Bell, and how to ring it.

.....

SMITH-CORONA PRINTER

By Paul W. Kittle
P.O. Box 1285
Loma Linda, CA 92354

At last I can print a decent letter, and this may be of interest to the MUG. I am responding to the call for comments on any new equipment used by MUG users for evaluation. The text you are reading was printed out on the new "Smith-Corona TP-1" daisy wheel printer. It sells for under \$900 (I paid \$799 + cable for mine due to an ad typo!). The cheapest I have yet seen it for is from the company I bought mine from, "Micro-printer Marketing" (800-523-9859) for \$845.

The good news about this new printer, aside from its great looks, is obviously the price. For the poor among us, \$845 is a far cry from \$2500+ for some of the more well-known brands of daisy-wheel printers. It is available in either 12 or 10 pitch (pica or elite) and has film ribbons for crisp letters. In addition, it can be purchased either as a serial or parallel port-driven printer. It feeds the single sheets of paper well (the tractor-feed is not yet available for fan-fold paper, I'll let you know about that when it comes out) and is just over-all a super printer for the bucks.

Now the bad part! Whereas this guy costs one-third the price of a Diablo or their kin, it also takes 3 times as long to print. Approx. 10 c.p.s. For those of you used to a dot matrix that zips along at 30 to 50 cps, it will seem very slow. For me, I have it set at 110 baud, and although it is adjustable, found that it couldn't handle anything faster.

That is the only real problem I have found with it so far, and I have dumped out several hours (straight) of term-papers on this guy. It doesn't seem to overheat on long runs, and just keeps plugging at it.

So, if someone is looking for letter quality, and can't afford an expensive printer, (and doesn't mind the printer taking about the same amount of time as a fast human typist), this is a way to go. Ann at Micro-printer Marketing is keeping me posted when other type-styles are available, so far they have 10 or 12.

Ed Note: Well, I read it, and it looked good - just like what you readers are now viewing, which is printed by a Diablo. However, speed is not the only sacrifice. It doesn't do super and subscript or micro-spacing. I'm not knocking it. It's just that potential buyers should check the total capabilities against their requirements.

.....

SULKY - THE HARNESS HANDICAPPER

by Bonjoel Enterprises

At harness tracks all over the country, people are enjoying the thrill of watching the horse they select to win cross the finish line ahead of the others. Unfortunately, there are not too many people knowledgeable of racing that can make consistent, accurate selections.

SULKY was developed for both the novice as well as the experienced handicapper. It takes past performance and driver data (available from various sources), for each horse in the race and forms a prediction of its outcome. All input data is written to disk so races can be calculated individually or for the entire racing day at each track. This provides a permanent record for future study.

By using this powerful program, the novice handicapper can achieve the accuracy formerly attained by only those of much experience. The experienced handicapper will use this as an adjunct (or replacement) for his normal handicapping method.

SULKY is supplied with complete instruction manual. Its clearly written text is designed to guide you through the few steps required in using the program effectively.

SULKY is designed to run on a minimum 56K, single-drive system with or without an 80-column printer, and requires CP/M for operation. It is presently available on Micropolis MOD I or MOD II media (CP/M), and on 8-inch, IBM format single and double density disk. SULKY (Disk with Instruction Manual) lists for \$125.

.....

```
*****
*
*      SULKY - The Harness Handicapper
*      from DAMAN
*      (205) 883-8113 or (205) 881-1697
*      Suite 14, 3322 S. Memorial Pkwy
*      Huntsville AL 35801
*
*      $108.75 Post Paid to North America
*      VISA and Master Card Accepted, COD available
*      or
*      $103.86 with cash discount
*      (cash, check, money order)
*
* Add $7 for airmail shipment outside N. America
* Add $5 for MOD I download
*****
```

MICROPOLIS MAINTENANCE

Fall is here, or at least my kids are back in school. Although the weather in Alabama is anything but cool, September tends to make me think it's time to get the computer system reorganized. That means checking out the location of all my programs, doing disk copies, getting documentation up-to-date, and tuning up the hardware.

That brings me to the subject of Micropolis Maintenance. Three items in DAMAN's inventory might be of interest to you: The Disk Maintenance Manual, the Diagnostic Disk, and the Alignment disk.

The Disk Maintenance Manual is two inches thick, covers all MOD I, II, III, and IV drives and all controllers which command those drives. Logic diagrams, schematics, and theory of operation are presented. Disassembly/assembly, trouble-shooting, and maintenance are fully discussed. Included in the Manual are procedures for using the Diagnostic and Alignment Disks.

The Diagnostic Disk is a program which responds to operator commands. In addition, it has some fixed sequences which ease the job of maintenance. Sequences are provided for test and adjustment of Drive Motor Speed, Instantaneous Speed Variation, Position Step Timing, Read Amplifier Gain, Head Compliance, Circumferential/Azimuth Alignment, Radial Alignment, Track Zero Switch and Stop, and Positioner Mechanical Adjustment.

The Alignment Disk is a precision recording of an elliptical track at location 16/36 - it works on both MOD I & II. The Alignment Disk is required for set-up of Circumferential/Azimuth Alignment and Radial Alignment.

Many of the test and adjustment routines require a voltmeter and/or an oscilloscope, though some, such as Drive Motor Speed, Position Mechanical Adjustment, Write Protect Switch, and Clamp Support Plate, don't.

```
*****
*
*      MICROPOLIS MAINTENANCE AIDS
*
*      from DAMAN
*      (205) 883-8113 or (205) 881-1697
*      Suite 14, 3322 S. Memorial Pkwy
*      Huntsville AL 35801
*
*      Maintenance Manual      $51.50   $49.18
*      Diagnostic Disk          49.50    47.27
*      Alignment Disk           49.50    47.27
*
*      All three together      135.00   129.00
*
* Above prices are postpaid to N. America. VISA
* and Master Card accepted. "CASH" is cash dis-
* count price (cash, check, money order). For
* airmail shipment outside N. America, add $12
* for Manual, $5 each disk, or $15 for the com-
* bination package.
*****
```

MICROPOLIS DISK CONTROLLER ENHANCEMENT KIT

The Shaw Laboratories kit contains the instructions and parts necessary to add several useful functions to your Micropolis disk controller board. Though the kit is not difficult, the work should be performed by someone who has some experience soldering integrated circuits to printed circuit boards and who is familiar with the orientation of the pins on integrated circuits.

The circuit modifications allow software control of the disk motors and allow the disk controller to generate *PHANTOM and/or to be phantomed. Phantoming is the procedure by which the disk controller can make other memory at the same memory location "disappear", or vice-versa.

MOTOR CONTROL

The ability to turn off the motors in the disk drives can greatly reduce the wear and tear the diskettes and motors receive, in addition to much quieter operation. The motors in the disk drives typically last through a couple of years of use and cost about \$60.00 each (including labor) to replace. Depending upon your location and the availability of service persons and motors, the down time you could expect can vary anywhere from less than 1 hour to many weeks. For a couple of dollars and a few minutes of your time, your disk drive

motors could conceivably last the life of your computer!

The circuit to allow software control of the disk motors requires several additions to the disk controller board. It requires that one integrated circuit (supplied) be installed in a spare location of the board, that a few traces be cut and that several jumpers be added to the back and front of the board. Software commands (in BASIC, FORTH, and Assembly) are explained.

PHANTOMING

Many times it is necessary to have memory mapped peripheral devices (such as disk drive controllers and video terminal boards) at the same memory spaces as system memory. That is, the system memory and the peripheral device addresses overlap. This cannot be allowed because both devices will normally try to respond when addressed, thus causing a conflict on the data bus. Additionally, many memory boards now have large capacities (32K to 64K or more) and do not have any way of disabling a small section of the memory address space to allow devices such as disk controllers to reside there. The usual way around this problem is through the use of the *PHANTOM signal line on the S100 bus. This signal allows a device to request (by generating *PHANTOM) that system memory or other memory mapped boards not respond when the address range of the device is accessed. In this manner, any device which will respond to *PHANTOM can be made to "disappear". For system memory, this removes the need for the ability for it to be disabled in small segments. For other devices, this could allow several devices to reside in the same memory space and be selected under program control. With the proper software and external hardware, this can also be implemented with the Micropolis disk controller.

Depending upon the phantoming requirements, different changes are made to the disk controller board. In the simplest case, where the disk controller needs to reside in the same space as system memory (or another device), just a few jumpers need to be added to the disk controller board. The parts required for this are already on the board. For a more sophisticated system where the controller is to generate *PHANTOM only under program control, or the controller is to be phantomed by another device or under program control, a single integrated circuit must be installed (unused portions of the same chip for motor supplied control are used) and a few traces must be cut and jumpers added to the back of the board.

All *PHANTOM generation by the disk controller assumes that the system memory (or other device, as the case may be) will respond to *PHANTOM. It is the user's responsibility to ensure that the remainder of the computer system will not be affected by the generation or use of the *PHANTOM line by the disk controller. Shaw Laboratories and DAMAN can take no responsibility for the use of the circuits in this kit.

Kit contains the required IC and 14 pages of instructions.

```
*****
*
*      MICROPOLIS DISK CONTROLLER ENHANCEMENT KIT
*      available from
*      DAMAN
*      (205) 883-8113 or (205) 881-1697
*      Suite 14, 3322 S. Memorial Pkwy
*      Huntsville AL 35801
*
*      $21.50 Post Paid to North America
*      VISA and Master Card Accepted, COD available
*      or
*      $20.00 with cash discount
*      (cash, check, money order)
*
* Add $4 for airmail shipment outside N. America
*****
```

EDIT/S: EDITOR FOR MICROPOLIS

by A.C. Lynn Zelmer
International Communications Institute
Box 8268, Station F
EDMONTON, AB T6H 4P1, CANADA
Copyright (c) 1981

According to the crystal ball gazers, the number of micro-computers in small business use will increase dramatically over the next few years. Unfortunately the software programmer seems to think that business applications start and stop with the finance department. Many of us will never buy a 'general ledger' and the various 'spread-sheets' (Visicalc et al) will have to be much more WORD oriented before we invest in that direction. There are good reasons why many small offices have invested in computer systems that perform word processing functions.

Dedicated word processors are purchased because the user wants the convenience of simplified text entry, with the text displayed on the screen exactly the same way as it will appear on paper. Micro editors are often primitive, lacking features found on both mainframe computer editors and word processors. Producing neat letters and reports usually requires the inclusion of special formatting commands in any text typed into the editor. These commands are processed by a text formatting program that produces the formatted text on the printer or console. Incidentally, this seems to be the most common text output system on mainframe computers and is similar to the system used by many computer controlled typesetters. The microcomputer owner has traded convenience for flexibility.

International Communications Institute is a small educational consulting practice with much of our work being overseas in developing countries. When we looked at computerizing our office in 1978 we had three major needs, simplified preparation of reports, maintenance of several medium sized data bases with 500 to 5000 items each, and communication with larger computers capable of supporting sophisticated statistical analysis techniques, etc. Also, we felt that we couldn't afford either a mini-computer or a dedicated word processor, the computerization was to help keep our overhead low.

After some months of investigation we settled upon a 48K North Star Horizon micro with dual Mod II Micropolis disk drives (double density), a 300 bps modem for communicating over the telephone, and an LA36 dot matrix printer. Our initial software consisted of:

- Organic Software's TEXTWRITER (TW),
- Creative Computer Associates' DATA MANAGEMENT SYSTEM (DMS),
- a communication program which allowed us to transfer files back and forth between our system and most mainframes that was custom written by TJB Microsystems of Edmonton, and
- the Micropolis operating system with LINEEDIT, BASIC, etc.

We had traded convenience for flexibility and minimal cost.

LINEEDIT is a simple, reasonably bug-free editor. As its name implies, LINEEDIT is line oriented. A line number is required at the beginning of each line, and changes are normally made one line at a time. LINEEDIT does have the facility of making global changes, but the CHANGE function operates upon the first occurrence in the line only. LINEEDIT does not have the ability to copy lines to another area of the edit buffer, and cannot move lines. It is possible to change the order of text segments by SAVEing the segments in separate disk files. The segments are then reordered using the APPEND and RENUM commands, but this is quite inconvenient and prone to human error. Perhaps most bothersome is LINEEDIT's need for commands to be typed out in full. Even a common command such as LIST can't be abbreviated, and every line change

needs the full EDIT command before the line number (individual character change commands within a line are abbreviated). LINEEDIT worked, it was saving us money compared to the use of the mainframe editor, with TW it gave us formatted text, but it wasn't as convenient as we would have liked.

We were quite intrigued then by an announcement from System/z, Inc. for their EDIT/S Text Editor. EDIT/S promised to provide LINEEDIT compatible files, thus it would work with our existing TW formatting program, as well as with our communications routines. Best of all, EDIT/S promised the ability to copy and move lines from one section of the text to another and had an 'auto-wrap' option for simplifying text entry. According to the advertisement EDIT/S did everything that LINEEDIT could do, only better, and at a price that seemed very reasonable (US\$ 45.00). Products don't always deliver what they promise, but this looked too good to ignore, so we ordered a copy.

I'm happy to report that EDIT/S delivers as promised, and adds a very welcome software tool to this Micropolis owner's toolkit. EDIT/S comes on a Micropolis MDOS Rev. 4.0 disk, MOD I or MOD II, and includes a concise, well-written operating manual. EDIT/S overlays the normal MDOS disk operating system, and thus has complete control over all aspects of the system (unlike LINEEDIT which could be affected by custom I/O routines, etc.).

OPERATING WITH EDIT/S

EDIT/S comes with a 50 page user's manual, and as mentioned above, the manual is concise and easy to read. The original Micropolis documentation for LINEEDIT had to be rewritten before my non-technical operators could use the system; the System/z manual is far superior. It includes a general overview and listing of the effect of each command. The Table of Contents functions as an index, and a complete listing of the commands appears in a two page summary.

Operation requires loading the Micropolis Disk Operating System (MDOS) in the normal manner, then typing
EDIT/S<RET>

EDIT/S takes a few seconds to load, it seems to be a multipart activity, probably because of the necessity to also load the custom system configuration. Sign-on includes the usual copyright information, etc. and the reminder that 'For assistance type HELP'. Typing HELP will display a summary of all the system commands. This summary is more detailed than the Command Format Summary, but does not include the operating examples contained in the manual. The 'on-line' summary requires a disk access, thus this facility doesn't take up valuable edit buffer space (32415 bytes on a 48 K system).

The CONFIG command requires the largest entry in the manual, and can be used to configure EDIT/S to any custom operating environment. CONFIG sets (or resets) the number of columns for your terminal, the default sizes for your printer page, a sequence of printer commands for software controlled printers, computer memory size, various terminal functions (including the column for automatic auto-wrap), and the default speed for listing the file on your terminal. CONFIG need only be run once, as the information is saved on the EDIT/S disk, but can be run again for subsequent changes.

EDIT/S commands consist of the command word followed by up to three parameters. Parameters, if required, must be separated by either a space or a comma. The backspace, properly implemented to erase the character being removed, is supported. Control X cancels the current entry line, Control C cancels the current command in operation when safe to do so.

Linenumbers are required with EDIT/S just as they are with LINEEDIT. Users familiar with the LINEEDIT automatic line number generation will find the same system available here. This requires

terminating each input line with a carriage return <RET>, and pressing the space bar to generate the new line number. In addition, EDIT/S automatically generates a new line using the auto-wrap feature. Using CONFIG or the temporary WRAP command the user sets the auto-wrap column. Text entry can be continuous, with EDIT/S generating a new line each time the cursor goes beyond the auto-wrap column and the space bar is pressed. It takes some experimenting to find the column most appropriate for an individual user's typing preferences, but once set the feature is very convenient. Lines can be renumbered, and edited (more about this later).

Entering the DOS command reboots MDOS and terminates the editing session (again more about this later).

EDITING WITH EDIT/S

Text entry is extremely simple with the use of the auto-increment function. A new file is begun by typing 'AUTO', 'A', or 'a', the editor responds with a line number, and typing can begin. Since I like to print drafts on normal width paper without using the TW formatting program I have the right margin set at about 62 characters. As soon as I press the space bar after the cursor reaches the 62nd column a new line number is generated. I have occasionally lost the first letter of a word as the cursor moves back, but since I type with two fingers this is not often. Pressing <RETURN> also generates a new line number.

Both automatic line entry modes provide a warning when the new line will replace an existing line. Typing Control X deletes the current line, Control C terminates the auto entry, allowing listing of the file, editing lines, etc.

The edit command can also be abbreviated; just type 'E' followed by the appropriate line number. At this point editing becomes character oriented, you can search for and replace any character or group of characters. In this regard EDIT/S acts much like LINEEDIT, except that there are some new commands; K deletes everything to the end of the line (Z deletes to a specified character, acting like K in LINEEDIT), R allows the typist to replace a string of characters, and A moves the cursor to the end of the line for appending new characters.

Global changes can be absolute or conditional. It is wise to use SEARCH before making an unconditional change, this will display all occurrences, allowing you to make a conditional change if necessary. A variation of the search function will even list all occurrences to the system printer. CCHANGE displays occurrence of the specified string and allows the operator the option of replacing the string. In this way it would be possible to guard against accidental changes to text as well as be selective in changing only certain occurrences. CHANGE initiates the replacement of every occurrence of the specified string.

Block movement of text is possible with the disk commands discussed above. To facilitate 'cut and paste' operations the operator has the MOVE and COPY commands. These two commands, used with RENUM (to renumber the lines), are as functional as any that I have used on a mainframe computer, and get well used on my system. For example, they allow you to quickly prepare summary statements; simply COPY the relevant paragraphs from the text and selectively eliminate sentences and words as necessary to make the new unit flow smoothly. Likewise a sentence or paragraph out of place can easily be relocated with the MOVE command. Because these commands work on a sequence of lines I always begin a sentence on a new line, it makes editing much simpler.

DISK OPERATIONS

Existing disk files can be edited with the LOAD command, and an edited file can be written to disk with SAVE. BLOAD and BSAVE allow selective reading

and writing, operating only upon a specified set of line numbers. MERGE allows a second file to be merged with the current file, and where duplicate line numbers exist, the disk file takes precedence over the file in memory. INSERT places a disk file into the buffer at a specific point, APPEND adds the file to the end of the current buffer.

Normal disk reads are very fast, disk writes subjectively seem to take slightly longer than with LINEEDIT, but not sufficiently longer to complain about. BLOAD is a command which allows the loading of a partial file, and is a slow operation. This feature is very useful when loading a file that is too large for the available file buffer and I'm willing to sacrifice some speed here just to be able to salvage a file that is too large. Previously I had to transmit the whole file at 300 bps to my main-frame computer, break the file into pieces, and return the individual pieces. For a large file this could literally take several hours.

Sometimes however, the disk operations are too fast and unforgiving. I do not like leaving disks engaged in the disk drives, especially when an editing session may last for several hours. My normal habit is to load programs and data, disengage the drives, and commence editing. As required I will re-engage the drives and load or save data as appropriate. MDOS is very forgiving in this regard and will always retry the head seek sequence long enough for the drives to come up to speed, even if I have forgotten to engage the drive before initiating the disk command. EDIT/S doesn't forgive, drive accesses are immediate, and immediately produce an error message and BELL tone.

This is a minor hassle, however the DOS command also initiates a disk access. If the disk drive is not engaged and up to speed the command fails, hanging up the whole system. Correction is only possible by booting the system from the computer reset switch. The DOS command is frequently used with LINEEDIT to allow a temporary exit from the editing program to check filenames, change file types, change system output parameters, etc. This operator was quite used to issuing the command without checking the status of the disk drive as I manipulated the file in memory, and old habits die hard. For compatibility, if for no other reason, I would have preferred that EDIT/S issue an error message rather than totally dying when I make this simple error.

MISCELLANEOUS UTILITIES

No editor is complete without the many little commands that make the life of the typist easier. EDIT/S won't allow me to perform any calculator functions while I am in the editor, but it will convert numbers between various formats (HEX to BINARY, etc.). This will be useful for assembly language programmers, but I find the ability to print a formatted copy of the text file more useful. PRF (or PRINTF) sends a paginated copy of the text to the printer complete with a short title and page numbers. This is invaluable for quickly generating short letters and reports that don't need the formal structure of justified text and the other features of properly formatted output. This feature alone paid for the cost of EDIT/S within the first two weeks of its use.

There are a number of other convenient utilities, it is possible to display the disk directory (complete with information on the number of available tracks), to scratch disk files and to change their file attributes. For some reason it is not possible to rename a disk file, I find myself getting quite inventive naming files so that I know absolutely which file is the current file and which is the back-up. Other features allow the control of compatible printers from the console, perhaps changing line spacing or moving to the top of a form. In addition, most commands can be in either upper or lower case letters.

SUMMARY

To summarize, EDIT/S delivered everything that System/z promised in their advertisements. EDIT/S is not perfect, but it is a good line-oriented editor with the capability to make block movement of text, conditional replacement of character strings, and informally formatted printer output. It has cut my text entry time significantly, giving me more time for concentrating upon the content of my reports. At a cost of only \$45.00 it represents a good investment for the users of the Micropolis disk operating system.

SUMMARY OF EDIT/S COMMANDS

APPEND - Loads a text file from disk and appends it to an existing file in memory.

ATTRS - Changes the file type of a disk file.

AUTO - Automatically generates line numbers for text entry. *

AUTOINCR - Sets the line number increment for the alternate entry mode (similar to LINEEDIT's AUTO mode). *

BIN - Math function to convert a number to its binary equivalent.

BLOAD - Loads a specified block of text from a disk file into memory.

BSAVE - Writes a specified portion of the text buffer to disk.

CCHANGE - Conditionally replaces every occurrence of a specified string of characters with a second specified string. Each occurrence is displayed, the operator must approve each change individually. *

CHANGE - Unconditionally replaces a specified string of characters with a second specified string. *

CLEAR - Deletes the text file buffer (clears the memory).

CONFIG - Used to initially configure EDIT/S for your system, or to change system default values.

COPY - Copies a specified block of text to another location in the text buffer.

DEC - Math function to convert a radix number to its decimal equivalent.

DELETE - Deletes specified text lines. *

DIR - Displays the diskette directory. *

DIRP - Prints the diskette directory on the system printer. *

DOS - Immediately transfers control to the MDOS operating system.

EDIT - Edits a specified text line. Has a number of character-oriented editing commands. *

Advance, Backspace, Abort, Append, Change, Delete, End Edit, Insert, Kill (to the end of the line), List, Quit, Replace, Search, Zap (to a specified character).

FF - Moves the printer paper to 'top-of-form' position.

FILE - Displays the file name, edit buffer addresses, and free memory available. *

FILES - Displays the diskette directory. *

FILESF - Prints the diskette directory on the system printer. *

FREE - Displays the number of available tracks on

the specified diskette.

HEADER - Specifies a header message for file listings. *

HELP - Displays a summary of the EDIT/S commands.

HEX - Math function to convert a number to its HEX equivalent.

INSERT - Loads a text file on disk into a specified point in the text currently in memory. *

LIST - Lists all or part of the file on the console. *

LISTP - Lists all or part of the file on the printer. *

LOAD - Loads a text file on disk into the edit buffer.

MERGE - Merges a text file on disk into the current file in memory.

MOVE - Moves a block of text from one location in the edit buffer to another location.

PAGESIZE - Specifies a new printer pagesize. *

PRINT - Displays the specified text lines on the console, ignoring TAB settings and omitting line numbers. *

PRINTF - Prints the specified text lines on the printer, ignoring TAB settings and omitting line numbers. Pages will be properly paginated, with page numbers and page header. Long lines will be 'wrapped' and included in the line count. *

PRINTP - Prints the specified text lines on the printer, ignoring TAB settings and omitting line numbers. *

PRINTER, PRINTER1, PRINTER2 - Outputs the specified user-defined control codes to the printer (specified with the CONFIG command).

RENUM - Renumbers the text buffer line numbers.

RESAVE - Save the text in the edit buffer in an existing disk file. Writes over the existing disk file.

SAVE - Saves the text in the edit buffer in a disk file.

SCRATCH - Deletes a specified disk file.

SEARCH - Locates and displays all occurrences of a specified text string. *

SEARCHP - Locates and prints all occurrences of a specified text string on the printer. *

TAB - Sets the TAB positions for the LIST and LISTP commands.

TABS - Displays the current TAB settings.

TITLE - Create, change or display the FILENAME for the text in the edit buffer. *

TYPE - Changes the file type of a disk file.

WRAP - To reset or display the 'auto-wrap' column.

* COMMANDS CAN BE ABBREVIATED

.....

 *
 * SYSTEM/Z'S EDIT/S - LIST PRICE \$45 *
 * On Sale until 10/15/82 *
 * from DAMAN *
 * (205) 883-8113 or (205) 881-1697 *
 * Suite 14, 3322 S. Memorial Pkwy *
 * Huntsville AL 35801 *
 *
 * \$37.50 Post Paid to North America *
 * VISA and Master Card Accepted, COD available *
 * or *
 * \$35.00 with cash discount *
 * (cash, check, money order) *
 *
 * Add \$7 for airmail shipment outside N. America *

MUG RESPONSE

A great many of you took time to write comments on the MUG when you renewed your membership. What follows is a partial "representative sampling" of the responses. More next month.

What I'd like to see in the MUG is more explanation on the software you are selling. The article on Bookkeeping was excellent. Why spend a lot of money until you know what the program will do. How about something on upgrading my Sol to a 24 by 80 screen. Quite frankly, I'd like my Sol to play Pac Man if possible. What about this? Anything in hardware would be great along with the latest software reviews. Time is short for doing much of my own software anymore.

Dick Peters, Oakland, CA

Dick: Recent issues of the SOL users group newsletter addressed upgrades of the SOL. You might wish to subscribe to PROTEUS at 1690 Woodside Rd., Suite 219, Redwood City CA 94061. One company that makes a 24 x 80 mod is Micro Complex, at 25651 Minos Street, Mission Viejo CA 92691 (714) 770-2168. Several people in the group have installed the mod. Some are happy, some aren't.

Although I have enjoyed reading about the other languages and other subjects I feel that our stated objectives were to stay with Micropolis Basic, RES and MDOS. There are at least five books out on CP/M, an untold number on Microsoft Basic and other groups and support for Pascal, Forth and Unix. It is for these reasons that I think we should stay with our stated objectives, and stay with Micropolis. Too much fragmentation and nothing gets the attention it deserves or is needed by the members.

A sharing of favorite routines might be a good place to start. A couple that come to mind are a routine for centering headings, finding the true length of a string, padding a string to a declared length. If every member submitted just one we would have a Micropolis tutorial. It could even be sold in booklet form.

Gene Riding, San Diego CA

I like what you are doing now quite a lot and think that you should continue as you are doing. I really wouldn't want to include very many hardware oriented articles, however, unless they apply directly to Micropolis hardware. I would like to see articles using FORTH and articles using good assembly language programming techniques. Perhaps a good 8080 sort routine with multi-levels and options. I am particularly interested in utility programs to manipulate data and files. Perhaps a 'small' data base system which would handle general purpose files with queries and reports (like

Apple's Filecabinet program) only in assembly language for speed. Since these are relatively large tasks, maybe they could be serialized over several issues.

John Barron, Austin TX

Your library disks may be of interest to me. But, the newsletter left me very confused as to the programs on specific disks.

Charles Seth, Fairborn OH

Charles: Your question was beyond a doubt the most frequently asked. The easiest way to get the full contents of the library is to look at the order from on Page 9. Order the library catalog of interest. The reason I haven't published the contents lately is two-fold. One, I just got everything organized again myself. Secondly, the MDOS Library listing runs 18 pages, and the CP/M Library listing is 8 pages. It would take two issues of the newsletter to print all this. While I intend to publish updates and additions, I probably won't publish the whole thing again. I can be overruled, of course. However, here's a list of the categories on the disks.

MUG MDOS LIBRARY: (1) Miscellaneous; (2) Miscellaneous; (3) Games; (4) Membership List; (5) S/W Vendors List; (6) System Utilities including LINE-EDIT disassembly; (7) Miscellaneous; (8) Business & Games; (9) System Utilities including Batchcopy; (10) MDOS Catalog System; (11) System Unique; (12) Technical & Household; (13) Newsletter Index; (14) Miscellaneous; (15) Z80 Assembler; (16) System Patches & Enhancements; (17) MDOS Disassembly; (18) System Disassemblies & Documentation; (19) Games; (20) FORTH; (21) System Unique; (22) Ham & Communications, though it's not ready; (23) Business; (24) Basic Utilities; (25) Education; and (26) Home.

MUG CP/M LIBRARY: (1) MBASIC Games; (2) System Utilities; (3) MBASIC Games; (4) System Utilities; (5) System Utilities; (6) System Utilities; (7) CBASIC programs; (8) Hardware Interfaces; (9) System Utilities; (10) Adventure Game; (11) Packman Source; (12) Packman & other Games; (13) FORTH; (14) MBASIC Graphics; and (15) BASIC-E programs.

My particular interest in MUG is tips for better programming, documentation of the Micropolis system, source of software available for Micropolis systems - in that order.

James H. Farish, San Diego CA

My own preference is for information about MDOS, MpBASIC and CP/M which will help to improve their use. MpBASIC has some very good features but I think it is badly let down by single letter names for constants, strings and string arrays, etc. This is particularly noticeable when you have been used to Microsoft Basic.

Frank Bamforth, Oldham, Lancs. England

There are a lot of other sources of CP/M information. I don't think that MUG is the place for more, nor do I think that we can have an effective newsletter if it becomes a CP/M newsletter. I have no problems with small amounts of information now being discussed, nor any problem with Advertisements for CP/M in Micropolis format.

As far as hardware reviews go, most of us have our basic systems, however we might discuss peripheral use. Most of the popular printers, terminals, modems, etc. have been reviewed in the 'slicks'.

SOFTWARE is the real reason most of us joined, and is the area that has been neglected by nearly everyone. I think that the programming tips are

SUPER, even those that are at a beginner's level. I would like to see more reviews of commercial software.

Inovative ideas will stimulate member interest in contributing to the newsletter. One such idea might be to each month print a listing of a program either from a library disk or a new program and offer a disk (or other prize) to the member who best cleans up that program. Or perhaps a prize for the member who submits the best program or programming hint each issue.

William Booth, Cordova AK.

William: Your prize idea is good - except I don't have time to "judge" them. Even if I did have time, I don't have the expertise in all the fields in which you members are interested. We'd need a panel of people, not just me. What do you suggest for a prize for any contribution? The reduced price on library disks is already some reward, but perhaps I could do more.

As far as I am concerned, anything that helps explain how Micropolis works is fine with me. The last article by Burks Smith was excellent. There is a lot I don't understand about the system. As an example, I can't get an assembled version of RES to execute. Also the destructive backspace will not work.

So, anything having to do with assembly language programming is very interesting. I think I learn more from your newsletter than the manual.

Tell Burks to write more articles.

Hap Little, Clinton WI

Hap: Consider Burks told. I echo your sentiments, and am very appreciative of the effort Burks has put forth for the MUG.

The reason the RES and backspace won't work is that you have a Vector Graphic. VG moved some things around. One of these days we'll have to document that, as the same question was asked by several people. Since I just acquired a VG System B at the house, maybe I'll do some looking.

Highlights of the last 12 months are:

- 1) Your catalog programs - just super.
- 2) Singer's 'BATCHCOPY' 'RESTORE' 'MDOSPATCH'.
- 3) Control-P for RES by Carinan.
- 4) All of Burks A. Smith's articles.
- 5) Reading directory from BASIC.
- 6) Running MOD-1 and MOD-2 together.
- 7) Turning off drives between access.
- 8) Auto configuration.
- 9) Micropolis hardware and software news.

It's hard to advise what I'd like to see in the future. I wouldn't have known to ask for any of the above a year ago. I don't mind some software reviews as long as this doesn't turn out to be an 18 column catalog for 'DAMAN'...I'd like to see \$-100 hardware reviews. I can't support sub-set membership since I'd like to see even what I can't use at this time. I'd like some articles on how to use fast machine language sort routines on basic data statements and basic data files. Would like to know how to pass single and multiple associated variables back and forth. Some simple examples of the above would help. Maybe Joel Shapiro could explain how he sorts his database files. Maybe Burks Smith could explain this in an article. You're doing a great job and you should know that renewing this membership is at the top of my list.

Richard Rusczyk, St Petersburg FL

I've been pleased with the content of the MUG Newsletters. There are not many places to get info directly related to my system or to find software that is compatible. I have a good library of CP/M software but I often prefer MDOS and Microp. BASIC for my own programming. Keep up the system info and info on software availability sources.

Jerry Luger, Torrance CA

I think your coverage on Basic/S and Basic/Z was very good. It was of interest to me because I plan to get one or the other before too long now.

I always enjoy seeing program listings printed out. Most of those in magazines are not easily adaptable to my Micropolis but that isn't a problem when we get them in your MUG Newsletter. Of greatest interest are short, handy routines to do things that are of use to most programmers at one time or another but maybe not used a lot. One shortie you published earlier had to do with rounding numbers off when printing them.

Hardware reviews are welcome also. In this vein,

1. () \$ Free Commercial MDOS S/W Price List
2. () \$ Free Commercial CP/M S/W Price List
3. () \$ 1.00 Commercial MDOS S/W Catalog
4. () \$ 2.00 Commercial CP/M S/W Catalog
5. () \$ 1.00 MUG MDOS Library Catalog
6. () \$.50 MUG CP/M Library Catalog
7. () \$42.00 Scotch DD, Reinforced Hub Disks(10)
8. () \$18.00 Year 1 MUG Back Issues (12)
9. () \$18.00 Year 2 MUG Back Issues (12)
10. () \$18.00 Year 3 MUG Current Issues (12)
11. () \$ 2.00 Micropolis Drive Head Pads (2)
12. () \$ 5.00 Lub, for Micropolis Drives (tube)
13. () \$72.25 \$69.00* Micropolis (MDOS) Ver. 4.0
Incls Manual, Basic & Utils
14. () \$51.50 \$49.18* MDOS Ver. 4.0 Manual only
15. () \$51.50 \$49.18* Drive Maintenance Manual
16. () \$49.50 \$47.27* Diagnostic Disk
17. () \$49.50 \$47.27* Alignment Disk
18. () \$21.50 \$20.00* Controller Enhancement Kit

* Cash Discount - (Cash, Check or Money Order.)

Adjacent items are postpaid to North America. Elsewhere, (airmailed) add \$12 (each) for MDOS and Manuals, \$7 for MUG newsletters, \$1 (each) for Catalogs, lubricant, and pads, \$4 for Controller Enhancement Kit. VISA & Master Card accepted. U.S. funds only. Prices subject to change without notice.

If you have a comment on the newsletter, a question for the MUG, or would like information on a piece of software, just jot a note below and sent to return address shown on the reverse side.

it might be useful to have departments in your newsletter for CP/M, hardware, MDOS, etc.

Another thing I would like is a list of members. I would be interested in forming a Micropolis club if I knew people in my area that had them. Maybe I should try to find them on my own and get them to join MUG.

One thing that would help a lot would be for you to not break articles up when putting them in the newsletter. It is somewhat frustrating to find an article chopped in several pieces and a page including parts of four different articles. If this is necessary I am not aware of the reason. Maybe your computer could think up a solution.

David P. Hunter, Champaign IL

David: MUG Library disk 4 has the full membership list on it, except for those who specifically asked not to be listed. Many members wrote, asking a question similar to yours. That disk should solve the problem. It has name, address, phone, type of computer, type of printer, interests, and other tidbits.

Several members also commented on the break-up of articles. I'll cease doing it. And here I thought I was being so "professional".

WHAT'S IT ALL MEAN?

Not too surprisingly, members have varied interests. Some want CP/M only, some want more hardware, some no hardware. What was surprising was that not one person thought that breaking up the newsletter into separate publications was a good idea.

Most members seemed to feel that the newsletter was doing a good job. That's great to hear, and thank you all for your kind words.

One idea that ran through several letters was the concept of a group project to produce a database manager. I'll pursue the idea next month.

Published Monthly by the MUG
Subscription rates:
U.S., Canada, Mexico; \$18/year; Other, \$25/year

FIRST CLASS MAIL

MICROPOLIS USERS GROUP

Buzz Rudow, Editor
604 Springwood Circle
Huntsville AL 35803
(205) 881-1697

SYSTEM/Z'S BASIC/Z

for MDOS or CP/M

No Royalties - No Run-time Charges
Generates machine code (8080/Z80)
Sort Verb - 2000 elements in 2 seconds
Alpha-numeric Labels
Chain with Common
BCD Floating Point Math to 18 Digits
Editor Tests Syntax As You Type
Recursive, Multi-line, User Functions
Dynamically Dimension Arrays
Selectively Erase Arrays
Trace and Single-step debugging
Multi-tired Error Trapping
Cursor Addressing, Reverse and Blinking Video
Want to see how it works? Get DAMAN's Mail-
List program for \$30. BASIC/Z run-time
packages are included. Just want further info?
Call or write for free BASIC/Z brochure and
mini-manual.

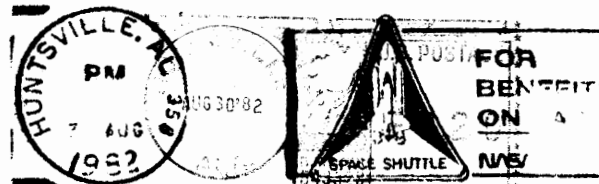
from DAMAN
(205) 883-8113 or (205) 881-1697
Suite 14, 3322 S. Memorial Pkwy
Huntsville AL 35801

\$291 Post Paid to North America
VISA and Master Card Accepted, COD available
or
\$278 with cash discount
(cash, check, money order)

Add \$7 for airmail shipment outside N. America
Specify MOD I/II, Apple, Osborne, NEC, N. Star

Full BASIC/Z Documentation \$35
(Credited to full order)

FIRST CLASS MAIL



FIRST CLASS MAIL