Don’t Ignore the High End

... or My Search for Manuscript Editing Paradise

by Carl Helmers

In this issue, we have included the first two of four users’ reviews of fully assembled and tested self-contained computers. The selection criterion is that the “typical user” (in two cases Dan Fylstra, one case Dennis Barbour, and one case myself) was able to purchase an off-the-shelf computer of the low end “appliance” variety. These machines represent a significant technological achievement in the form of a fairly inexpensive computer with a high level language (BASIC) and limited mass storage facility in the form of audio or modified audio cassette tape. In their class of price and performance circa December 1 1977, these “appliance” computers represent perhaps the ultimate we have seen to date.

As time goes on, I’ll update the ultimate, but that is not my purpose in this essay.

There is a utilitarian class of personal computer products which I have used most in my homebrew system. I refer of course to the high end of performance in small computer systems, the system with perhaps 16 K to 32 K memory, single or dual full size floppy disks, terminal device, and all the accoutrements of what we used to know as “minicomputers” but now within the price range of the family that can afford an $8000 car. My attention was turned to this class of machine recently when I began shopping around for some equipment and software to automate several aspects of the manuscript preparation process which I and my associates in the editorial department perform daily at BYTE.

Continued on page 136

Some Enticing Advance Words

Turtles are coming. In mid November 1977, Dan Hillis of the LOGO project at Massachusetts Institute of Technology called BYTE and posed one of those questions which have only one answer, “Would readers of BYTE magazine be interested in an inexpensive kit version of the famous LOGO project turtle robots?” Dan outlined on the phone the idea of a kit with a target price of $250, and suggested that I take a trip to the MIT AI (artificial intelligence) Laboratory to take a look at his prototypes. I was delighted to accept the invitation, which included a heady tour of some of the projects at the AI laboratory and the Architecture Machine project’s advanced color video display technology.

The turtle is not yet ready for market, but it is getting close, as can be seen by photo 1 which I took while lying prone on the floor of the laboratory as Dan put the prototype through its paces. The idea is to produce a rugged and childproof motion output device with tile drag plotting capabilities provided by a ball point pen controlled by solenoid. In the past turtle robots have proved to be excellent devices for teaching children programming concepts, both in simulations on screens, and as mechanically mobile output peripherals. The intent is to make the new turtle product virtually indestructible with Lexan plastic housing and wheels, and rugged mechanical and electronic innards. Dan demonstrated some of this indestructibility with a prototype sans housing by dropping it from a height of about two meters onto the carpeted floor of the laboratory.

Of course a turtle is not all hardware, so versions of the LOGO language adapted to personal computer architectures will be required. Dan and his associates (all quite young) at MIT have formed a company which intends to introduce turtles much more formally sometime in early 1978. We’ll expect to keep readers informed of this exciting prospect as it develops and gets closer to production.

Photo 1.
The urge to solve the problem of automated editing tools with a small system has been growing within me for some time. Prior to starting this publication, I had often used the text editor of IBM’s TSO software in preparation and testing of programs and job control in a large system timesharing environment. Earlier I had some experience with a more sophisticated editor, TECO on a DEC PDP-6. As a result of this experience, I dreamed of the possibility of a multiple window display, upper and lower case graphics, and lots of characters. Since then, as I found myself getting into word pushing with a vengeance as a daily way of life, the need for a practical approach to automated editing of manuscripts has become more important. I have seen examples of what a fast computer and good software can do in demonstrations of some automated editing tools used with PDP-10 hardware at Stanford Research Institute (now known as SRI International). Recently I have seen similar examples at Massachusetts Institute of Technology’s Artificial Intelligence Laboratory. Ah, the frustration mounts! When would I find the time or money to duplicate such tools?

In many respects, I feel like the proverbial “shoemaker’s child,” running around “barefoot” with respect to the very technology about which I am most concerned. To be sure, I am a little hypocritical when I make this comment, for I do have a quite complete experimental laboratory in my basement: it currently has a loose federation of four different microcomputer systems talking to each other with programmed IO through parallel ports, each system specializing in a particular phase of the music control problems I spend my spare time upon. But this homebrew system is not a production tool for daily use in the office. Its status is anything but frozen, and as I experiment with the system it begins to resemble a bar of butter on a hot summer day. I am loath to use it as a production tool, for its very charter as a test bed for experiments is inconsistent with using it for regular work. Our present concept of automation in the editorial office has consisted of an IBM Correcting Selectric II typewriter on each editor’s desk, the best electromechanical system, but far short of a computer text editing system.

Knowing full well that my personal homebrew system had limitations, I started a process of defining the requirements of a production text editing system which would fit my needs at the office and would be consistent with a “low” price tag using pro-
ducts drawn from the small systems field. It is in this activity of defining an editing system with as much standard hardware and software as I could get, that I found myself examining that class of complete systems characterized by extensive system software, extensive memory, video displays and mass storage on floppy disks. The activity of specifying such an editing system is the extent to which I have carried the search as of this writing in early December 1977. During 1978 I hope to see a practical fulfillment of this concept in one system. The result will not come up to the level of what I have seen at SRI in the way of high resolution displays and interactive text manipulation concepts, but I expect to get a workable and useful approximation at a price which I can afford. The requirements of the minimum system which will meet present goals are fairly simple to state:

- The editing software must have a sophisticated programmable nature allowing definition and execution of macrooperations (with multiple levels of conditional execution).
- The editing hardware and software must support upper and lower case video displays with enough capacity to allow multiple simultaneous "windows" of information about the article file being edited. Initially, a command language oriented to an ASCII keyboard will suffice for interaction.
- The editing system must support software of a good floppy disk operating system with all the usual utilities, a macroassembler and one or more compilers.
- The editing system must read, convert, edit, reconver and write IBM compatible EBCDIC text files on floppy disk media.

These requirements are based upon what I think I can get within the constraints of price (under $8000 to $10000 per system) and little or no software development or engineering design time on the part of myself and my associates at BYTE. Let's explore some of the reasons why these features might be important.

Programmable Editing...

As an individual thoroughly schooled in the uses and abuses of algorithms, one of my dearest wishes is to be able to specify procedures for the alteration of manuscript text, rather than mere "find" or "find and replace" or "delete" as found in most crude character oriented text editor programs.
LEARN TO PROGRAM MICROCOMPUTERS

And at an affordable price. The Modu-Learn™ home study course from Logical Services.
Now you can learn microcomputer programming in ten comprehensible lessons. At home. In your own time. At your own pace.
You learn to solve complex problems by breaking them down into easily programmed modules. Prepared by professional design engineers, the Modu-Learn™ course presents systematic software design techniques, structured program design, and practical examples from real 8080A micro-computer applications. All in a modular sequence of 10 lessons . . . more than 500 pages, bound into one practical notebook for easy reference.
You get diverse examples, problems, and solutions. With thorough background material on micro-computer architecture, hardware/software trade-offs, and useful reference tables. All for only $49.95.

For $49.95 you learn design techniques that make software work for you. Modu-Learn™ starts with the basics. Our problem-solution approach enables you to “graduate” as a programmer.
Circle the reader service number below to receive our free descriptive brochure and course outline.
Use your Master Charge or VISA card to order today. Call Pat at (415) 965-8365.

As noted earlier, I was spoiled at one time in my life when I had about six months exposure to Digital Equipment Corporation’s TEOC text editor on a PDP-6 at the University of Rochester, Rochester NY. That taste was enough to convey the power of an editor in which I could compose conditional text alteration procedures, and execute them in real time to edit my files. (The files I was editing were a number of FORTRAN programs for John Conway’s game of Life, implemented at the expense of my grades as a physics graduate student.)

For those unfamiliar with TEOC, which is still used in a number of forms on Digital Equipment Corporation’s computers, all I can say is that it is addictive. I had no such feelings for the IBM TSO editor which I used later in my experiences, or for the quickly implemented and quite crude text editor I have since written for my homebrew system. The TEOC I used six years ago has probably evolved considerably since I used it, but that one taste was enough to leave me in love with the method.

The idea of programmable character oriented text editing language is simple: the ability to program use of primitives for scanning text in search of matches to a syntax pattern, and conditionally modify the text based on local context. A classic example with which I deal all the time at a manual level is the case of a decimal number followed by zero or “n” blanks, then a double quote mark ("), or one or more variations of the abbreviations, and full spellings of the singular or plural form of the English language word “inch.” In Backus-Naur form, this syntax might be noted:

\[
\begin{align*}
\langle\text{inch-expression}\rangle &::= \langle\text{number}\rangle \langle\text{blanks}\rangle \langle\text{inch-units}\rangle \\
\langle\text{inch-units}\rangle &::= " \mid \text{inches} \mid \text{in} \mid \text{inch}
\end{align*}
\]

The object of the macro to be composed for this case is recognition of this syntax or approximations of it, and replacing each instance with a standard representation of the measurement which includes metric equivalents calculated from the original English measurement. Ignoring the semantics of the transformation, the resulting syntax should be:

\[
\begin{align*}
\langle\text{inch-cm-expression}\rangle &::= \langle\text{number}\rangle \langle\text{blanks}\rangle \langle\text{inch-units}\rangle \langle\text{metric-inches}\rangle \\
\langle\text{metric-inches}\rangle &::= (\langle\text{number}\rangle \text{ cm})
\end{align*}
\]

This “inches to centimeters” transformation

---

Circle 63 on inquiry card.

711 Stierlin Road
Mountain View, CA 94043
(415) 965-8365
is the prototype of a number of similar operations which convert any one of the usual English system units to metric within text. At present I do this manually with a pocket calculator featuring metric conversions, and a blue pencil for recording the results. With some ingenuity, a text editor of the sort I am looking for can probably handle most such conversions semiautomatically with interactive approval of the results during execution of the macro.

As frosting on the whole macro text editing capability, it would be most useful to record a library of symbolically referenced procedures such as "inches to centimeters" as they are developed, so that they can be called as needed through a simple sequence in the text editing command language. This may not be achievable through standard software which is used honoring the principle of "no software development" on my part.

Capacious Upper and Lower Case Displays...

An editing system for manuscript materials without upper and lower case facilities is a contradiction in terms as far as I am concerned. One can certainly put up with upper case only text editing hardware in the context of any one of a number of existing high level languages and assemblers. But this absence of upper and lower case capabilities in such systems' software and hardware is an archaic carry-over from the earlier days of data processing and computing, when keypunches and Teletypes dominated the technology. The job of manuscript editing is impossible without a full upper and lower case capability. In judging articles for the magazine, I often will not accept a manuscript from an author which is typed upper case only, unless one or more exceptional conditions exist.

The use of video displays rather than hard copy is also crucial. I have no intention of waiting for a slow mechanism and creating mounds of waste paper as a result of the editing process. To be sure, hard copy will be used, but only in the form of a confirmation printout of the original state of the file supplied after the act of keying a manuscript to disk, and possibly as a final review of the results of the technical editing. An electronic display is essential to allow quickly scanning and examining details of text without the speed limitations of mechanical displays. The displays must be capacious also, since a mere 12 lines by 80 characters, or 16 lines by 80 characters, does not give a big enough field.

The ideal goal, which I have seen in the systems at the artificial intelligence laboratories mentioned earlier, is at least one full page of text, where a "full page" means perhaps 50 to 60 lines of text. This is the rough equivalent of the capacity of a page of single spaced typewritten copy. This capacity is then allocated to multiple logical windows on the data using the system software of the text editor. Such a display does not exist in any inexpensive practical form for the small system field, although it can of course be simulated. There are several Altair (S-100) bus products on the market which put up 24 or 25 line displays of 80 char-

---

"CHEAP TALK" The $115.00 Modem from Phone 1 now comes fully packaged!

Printed circuit modem for customer mounting within terminal. Board dimensions: 4 1/2" x 5" x 3/4" ....................... $115.00

Packaged modem with above printed circuit in table top cabinet and DB-25S connector and 6' telephone line cord. (Specify termination for line cord. Either quarter-modular or spade lug). Cabinet dimensions: L.7 1/4" W.5 1/2" H.3 3/4" .................. $133.00

- Two week delivery - Full duplex
- 0-300 baud - 103 type modem

Call or write us today.

PHONE 1
1330 East State St.
Rockford, Illinois 61108
Phone 815/962-8927

Printed circuit modem from Phone 1 now comes fully packaged!
The concept of assigning logical “windows” on the file’s data is essential. In each article I edit, I have numerous categories of information which are logically separate, though related by belonging to the same entity. There is always the main body of text of the article; there is the captions list with subcategories based on the type of illustrative material being captioned; there is always a set of summary information such as the author’s name, “about the author,” the author’s address, etc. There are sometimes additional optional materials such as glossaries, boxes of text which are auxiliary arguments not part of the main text, etc. Often it is necessary to scan one such logical subset of the article while retaining a segment of another such subset in view. For example, when checking a glossary item, the main text might be scanned while the glossary is maintained on the display. Or, when making a change of phrasing to clarify an author’s point, the original text should remain on display while the modification is being made, just as I can always see the original text when I make pencil corrections in the manual method of editing. The ability to allocate multiple windows pointing to the same data set is only effective when the display area is big enough so that each window is of nontrivial size.

**IBM Compatible Media.**

The requirement of IBM compatible floppy disk media is the only one which is driven by an external consideration specific to the context of BYTE publications. If the editing software were being used by an individual in an isolated situation, there might not be a need for either IBM compatibility or full size floppy disks. But here at BYTE we have an existing system of production which employs equipment that is supplied by IBM which uses floppy disk media for temporary storage of manuscript information during the typesetting and production process which happens after the technical editing of articles. The cleanest interface to this existing system is to have the input keying operations performed prior to the technical editing, then physically transport one floppy disk along with the source documents of the article to the editorial offices. After the technical edit is completed, the floppy disk is then returned to the production department along
with the source documents, with all technical changes effected by the technical editors, rather than transcribed by production editors from penciled or typed corrections to a printed manuscript.

An Approach to Selecting Such a System's Hardware

With this rough functional description of what I want, how do I go about converting the idea into a working system with off-the-shelf hardware and software? This question is what caused me to begin a process of examining some of the more complete product offerings by various manufacturers which had the potential for meeting the required specification. The fact that I will even consider several of the former kit manufacturers for the personal computing marketplace as the principal suppliers is evidence of the maturity of this field as it has grown over the past few years.

My first and most important criterion is that the system must be purchased in complete form from one manufacturer (complete meaning "assembled, tested and demonstrated"). I do not want finger pointing (at least initially) due to conflicts over the ambiguous definitions of different manufacturers' hardware and software conventions. In my case, this will probably be accomplished by dealing directly with the manufacturer since we already have an established business relationship as a magazine; for readers of the magazine this can be accomplished by using a computer store as the local equivalent, with the proprietor selling a complete and working system demonstrated in the store before purchase. I may mix and match peripherals from other sources at a later time, but such

MORE 6800 SOLUTIONS:

FOR SWTPC 6800

AS-1 ANALOG INTERFACE MODULE
- FAST, ACCURATE A/D, D/A CONVERSIONS
- EIGHT ANALOG INPUT CHANNELS
- ONE HIGH-SPEED ANALOG OUTPUT
- OCCUPIES ONE I/O SLOT
- $67.50 KIT $115.00 ASSM.

SC-1 SERIAL INTERFACE CARD
- RS-232 SERIAL INTERFACE
- FULL CONTROL SIGNALS IMPLEMENTED
- MOST VERSATILE SERIAL I/O AVAILABLE
- $30.00 KIT $40.00 ASSM.

U.S. Orders Postpaid

FOR MOTOROLA "D2"

DA-1 UPGRADE KIT
- CONVERTS D2 TO TERMINAL I/O
- RETAINS FULL CASSETTE I/O CAPABILITY
- INCLUDES RT/68MX ROM MONITOR/O/S
- ALLOWS USE OF BASIC, EDITORS, ASSEMBLERS, ETC
- RS-232 SERIAL INTERFACE
- DA1 w/70 P. MANUAL $69.95
- MOTOROLA MM568104 16K RAM
- 16K BYTES DYNAMIC RAM
- ENGINEERED SPECIFICALLY FOR "D2"
- EXORCISOR - TYPE BUSS
- $395.00 ASSEMBLED

MASTER CHARGE & VISA Welcome

M6800 SOFTWARE

A/BASIC COMPILER
- 8K EXTENDED BASIC COMPILER
- CONVERTS BASIC PROGRAMS TO FAST, MEMORY EFFICIENT MACHINE LANGUAGE
- DESIGNED FOR RT/68 — SUPPORTS MULTIPROGRAMMING
- EXTREMELY VERSATILE I/O
- KC CASSETTE + MANUAL — $49.95
- RT/68MX OPERATING SYSTEM
- EXPANDED CONSOLE MONITOR ROM
- REAL TIME OPERATING SYSTEM
- DIRECT MIKBUG REPLACEMENT
- PIA OR MULTI-ACIA I/O
- RT/68MX ROM w/MANUAL $55.00

WRITE OR CALL FOR FREE CATALOG

MICROWAVE SYSTEMS CORPORATION
P.O. BOX 954 • DES MOINES, IOWA 50304 • (515) 279-9856

Circle 21 on inquiry card.
incremental additions or modifications will only be done with the backup of a known working configuration.

The second and equally important criterion is that the supplier must be able to supply a complete set of systems software, already developed and operational, at the time of delivery of the system. This software includes a disk operating system, the text editor framework for the manuscript editing operations, and the usual macroassembler, linking loader and compilers of a "minicomputer-like" small system.

I highly recommend that potential users of a system either purchase or examine the manuals of the system they intend to buy, going over the manuals for purposes of evaluating completeness and quality of hardware and software support. In the past month or so, two examples of excellent documentation in a small computer system have come to my attention. One example is the so-called "preliminary" (but not very "preliminary" by most standards of documentation) manual of the Apple-II computer, discussed in the review appearing separately in this issue. A second example is provided by a system I am examining as a potential candidate for the editing function: the Technical Design Laboratories' Xitan product which has been seen publicly at a number of shows and computer stores. I recently travelled to New Jersey to visit TDL at the invitation of Roger Amidon. During that day long visit, aside from the usual rituals of visiting a company manufacturing in this field, I had opportunity to examine much of their software in operation, and to examine their version of a high capacity 80 by 25 character display with upper and lower case graphics. I picked up a pile of approximately 3 inch thickness filled with the excellent manuals for their software, which includes a FORTRAN IV with nearly complete ANSI features plus extensions, a macroassembler, CP/M by Digital Research as an optional operating system, a data base management system for the FORTRAN programmer called "Micro-SEED," and the ZTEL text editing language which was the most important attraction of the trip. From the documentation, I was able to learn that much of the capability I would like to have for a text editing system used with manuscripts is likely to be available more or less off-the-shelf in ZTEL. There will be some custom work required, to be sure: for example, the editor does not support multiple editing windows, and will not be able to symbolically store debugged text editing macros on the floppy disk without some thought. But it may prove to be most of what I was looking for, especially with the use of two of the 80 character by 25 line display boards TDL is now manufacturing.

At the time of this writing, no decision has been made yet as to the details of such as editing system, nor have we purchased any equipment. I intend to review the manuals of a number of the manufacturers who offer good combinations of hardware and software which are potentially usable in this system concept before any final decision and purchase is made. But now that I have made the investment of reading the manuals, TDL seems to be a company which is well worth looking at.

![Alpha Z-80](image)

**ALPHA Z - 80**

*495* $ ASSEMBLED

- 12 SLOT MOTHER BOARD
- 12 CONNECTORS (S-100 BUS)
- 17 AMP POWER SUPPLY
- HEAVY DUTY CABINET WITH FAN
- Z-80 CPU BOARD: All Sockets Included, Gold Contact Fingers, High Quality Glass Epoxy PC Board, Double-Sided, Plated Through Holes, Requires Only +8 VDC / 800 MA.

**NOTE:** 22 SLOTS/30 AMP POWER SUPPLY - $595.

ADS also sells IMSAI and NORTH STAR Assembled at Kit Prices + LEAR SIEGLER ADM 3A Assembled at $888.00. TERMS: Cash with Order. Prices include Freight. (N.C. Residents add 4% Sales Tax.)

*Z-80 is a registered trademark of Xilog Inc.*

**ADS**

ALPHA DIGITAL SYSTEMS

Data Acquisition, Computation and Control

**ROUTE 4 BOX 171A**

BOONE, NORTH CAROLINA 28607
The theme which introduced this digression into editing hardware and software is that there are machines in existence right now, which are marketed as finished products, and which, with various degrees of software support, can be had at moderate prices in the $3000 to $8000 range through computer stores. This class of machines is what is needed to do “serious” personal computing. The characteristics of this class of machines, which has been maturing lately in the marketplace, might be given as: 16 K bytes of programmable memory and up, dual floppy disk drive, terminal, optionally some form of printer, and availability in a fully integrated form which needs no assembly from kits. Styles and configurations vary, of course, but the important point is that for the price of a good pickup truck, one gets a computer which can be effectively used for both utilitarian and frivolous computational uses at the whim of its owner. If the processor is an 8080 or Z-80, then Gary Kildall’s CP/M operating system sold by Digital Research is frequently used with this kind of system, and often there is both a compiled high level language (for example TDL’s or Cromemco’s FORTRAN IV compilers) and a macroassembler. Rumor has it that PASCAL compilers exist which will be available shortly for this type of disk based system.

Your MicroComputer can be a Money-Making Machine with SOFTWARE from SYNCHRO-SOUND

Here are two Software packages that enable you to profitably make full use of the potential in your small computer system: Synchro-Sound’s interactive business systems designed for use on an 8080- or Z80-based microcomputer with floppy disks and a hard-copy printer. Written in Microsoft Basic (known as Altair Basic*), the software supports file maintenance, query and reporting functions. This support includes creation, modification and deletion of customer records and processing of transactions to be posted against the files.

ACCOUNTS RECEIVABLE PACKAGE
The A/R module creates a history file of charges, credit and payment records for audit purposes. Payments can be applied to a particular invoice, or to the oldest open items. Batch Mode for monthly statements, aged receivables schedule, and delinquency notice as well as on-line inquiry mode for specific information are both fully implemented.

INVENTORY PACKAGE
The inventory system maintains a current listing of stock items featuring on-line transaction processing. Reporting functions include a master inventory listing with price and cost data, reorder alerts, vendor specification, and summary valuations. Year-to-date usage records are kept for each inventory item to facilitate sales analysis.

Operators Manual & Disk—Single Copy
Operators Manual only—$35.00 refundable upon purchase of Software Package.
Microsoft Disk Basic—Single Copy $350.00

SYNCHRO-SOUND Enterprises, Inc.
193-25 Jamaica Avenue
Jamaica, New York 11423
212/468-7067 TWX 710-582-5886

WRITE OR CALL FOR FURTHER INFORMATION
Additional Software Packages will be available in the near future

*Altair is a trademark of MITS, Inc.