

# SORCERER'S APPRENTICE<sup>TM</sup>

PAGE 81

VOLUME 3

NUMBER 5

JULY 15, 1981

INTERNATIONAL COMPUTER USERS GROUP/NEWSLETTER

-----  
Copyright (C) 1981 by Sorcerer's Apprentice

Price \$2.00  
-----

## IN THIS ISSUE -

DAISY WHEEL GRAPHICS.....	81
ODDS & ENDS.....	82
USE OF RESTORE.....	82
REALTIME PRINTER DRIVER.....	83
MX-80 PRINTER.....	84
HELP KEYING-IN.....	85
HARDWARE NOTES.....	86
WORD PROCESSING CORNER.....	87
TRS-80 PROGRAM CONVERSION...	88
PASCAL PORT.....	93
DISK NOTES.....	94
DIABLO SERIAL DRIVER.....	96
EVALUATIONS.....	97
DUSTINGS.....	98
4TH TIP.....	100
WORD PAC TIPS.....	103

## ----- LOW COST INTERFACE FOR DAISY WHEEL PRINTERS - by Don Ellis

It's not uncommon to find used Diablo Hytype I printers in the \$600 to \$800 price range. Occasionally, somebody is even lucky enough to find an older Qume for about this price. The prospect of daisy wheel performance at these prices is pretty attractive. Add to that the possibility of using small increment movements to get hard copy of Sorcerer graphics, and the proposition can become mouth-watering. The only problem is that it takes some hardware between the Sorcerer and the printer to make the system work. If you happen to have an S-100 box attached to your Sorcerer, there are some printer driver boards on the market that will do the job for less than \$200.

Since I didn't already have an S-100 box, I was horrified by the roughly \$600 price for the interface card plus a box for it to live in. Fortunately, there are less costly alternatives. The one I'm describing has proved to be a serviceable interface that can reside in a calculator-size box (such as PacTek's HP) and costs about \$30 in parts.

Since the daisy wheel printer requires more than 8 bits of parallel data, direct connection to the parallel port is **not** an alternative. It takes two bytes of output from the Sorcerer for each instruction to the printer (print, carriage movement or paper movement). One approach would be to use an external register in which printer control words are assembled from sequential data bytes from the Sorcerer. An easier (and less expensive) approach is to go through the 50 pin bus connector and use some of the address lines to transmit data.

A good plan is to use the high order address byte (address lines A8 through A15) for data and reserve the device address. This insures that printer data will never erroneously enable some other device such as the parallel port. It also assures that an address, which was supposed to select some other device, won't inadvertently operate the printer.

With the Z-80 **OUT (C),r** instructions, address lines A0 through A7 are loaded from the Z-80 **C** register, and address lines A8 through A15 are loaded from the Z-80 **B** register. So, the software implementation of this assignment of address lines is straightforward.

(continued on page 101)

## ODDS &amp; ENDS - by Ralph LaFlamme, Editor

With this issue, we have added two additional columns. The first is a FORTH column by Tim Huang and the second is a product review column by Emiliano De Laurentiis. A column by Dan Behmer, our International Editor, will appear at irregular intervals. Dr. Behmer is Exidy's Director of Computer Education and is presently on assignment in the far east. He will report on his travels upon his return in the fall. The Apprentice Port column returns in the next issue.

There is no need to explain the havoc inflation has been raising with everyone's costs. We are no exception. In order to help keep our costs in check, we've been trying to maintain the size of the Newsletter to 20 pages. Nonetheless, in order to accommodate the new columns and other articles we wanted to publish, we've found it necessary to increase the size of this issue to 24 pages. So, this is an inflated issue containing a 4-page bonus!

I would like to clarify a point I made in issue 3.3. I asked that the Newsletter not be reproduced for others. It is our intent to help those who need it whenever we can. If an article will help someone with a problem, and they have no other source of help, we certainly have no objection to a copy of that article being passed on to them. We do, however, object to the wholesale reproduction of an issue for friends, etc. We hope to increase the size of the Newsletter with the next volume and will certainly need these revenues to do so.

ERRATA: Bob Roth sends along a correction for his Z-80 Relocator program which appeared on page 78 of issue 3.4. Address 50EC reads as "ED" but should read "FD". He apologizes for any inconvenience caused by this error.

ERRATA: There was a missing parenthesis in the Basic program on page 79 of issue 3.4. The line in question should have read:

```
32000 AD = PEEK(-4096) + 256 * PEEK(-4095)-2 + 65536 * PEEK(-4095) > 127)
```

The deadline for submissions for the next issue is August 1st.

## USE OF RESTORE (&lt;nnnnn&gt;) COMMANDS - by Moe Comeau

Unless Exidy has updated their BASIC Manual, they have not documented a useful capability in their Microsoft BASIC. It is that you can use a LINE NUMBER following the RESTORE (Graphic^ ) command.

This can be handy if you don't know where you are in the data set due to a variable number of READs or, more likely, programmer confusion.

This can also be handy if a particular set of data is reread several times but the dumb programmer put it at the end of the data set instead of the beginning. (How many programs have you seen that have this construct: FOR I=1 TO N: READ A: NEXT?) This is a colossal waste of time just to position the Data pointer!

Try the following program:

```
10 DATA 1,2,3,4,5
20 DATA 6,7,8,9,0
30 RESTORE 20
40 READ I
50 PRINT I
60 END
```

RUN (Graphic \*)

6

READY

Note the Data pointer ignores the fact that line 10 had data that never got read.

Maurice Comeau, 20 Karrens Way, Downingtown, PA 19335

**BASIC REALTIME PRINTER DRIVER CONTROL - by Moe Comeau**

```

10 :      REM REALTIME PRINTER DRIVER CONTROL
20 :      REM OR, A POOR MAN'S LPRINT
30 REM The Sorcerer's SEND routine location is stored in two
40 REM bytes of the Monitor Work Area, MWA +3F. In the
50 REM 32K Model, this address is 7FD0 or decimal 32720.
60 :
70 REM My printer driver is located at 00C0 hex, so this is
80 REM the location I need to POKE into these bytes.
85 REM Note: C0 hex = 192 decimal
90 :
100 A$=""
110 FOR I=1 TO 10
120 PRINT I
130 POKE 32720,192
140 POKE 32721,0
150 PRINT A$
160 REM Now reset output to E9F0, the video out routine.
170 POKE 32720,240: POKE 32721,233: REM Same as >SET O=V
180 NeXT
190 :
200 REM The above program will print 1-10 on the screen
210 REM but only send 10 carriage returns to the printer.

```

Lines 130 and 140 are the BASIC equivalent of the Monitor command:  
SET O=C0

---

**mentzer**  
**electronics**

590 South Hill Boulevard, Daly City, California 94014  
(415) 584-3402

\*\*\*\*\*

SPELLBINDER - Word Processor	\$395.00
SPELLGUARD - Dictionary Program	\$295.00
EXIDY 1.1 Monitor ROMS	\$ 45.00
EXIDY S-100 I/O Card. Allows any S-100 box to be connected to the Exidy Computer. LIMITED QUANTITY	\$225.00
CP/M 2.2 with BIOS.ASM file. For use on Exidy with Micropolis hard sector disk ONLY (CP/M is a trade-mark of Digital Research)	\$190.00
SPECIAL BIOS      A WAY OUT OF THE EXIDY CP/M BLUES! Make your system compatible with standard Micropolis CP/M formats. Drive A = CP/M 2.2 format, Drive B = Exidy CP/M 1.4 for Micropolis hard sectored format. Allows you to PIP your old files to the new CP/M 2.2 format. Requires CP/M 2.2. WILL NOT TRANSFER CP/M 2.2 FILES TO THE EXIDY FORMAT.	\$ 50.00

\*\*\*\*\*

MASTER CHARGE and VISA on orders of \$50.00 or more  
Shipping will be added to all orders.  
California residents add 6% Sales Tax.

## USING THE MX-80 WITH THE SORCERER - by Frank Voss

As it stands, the Sorcerer computer is a pretty neat and versatile package. Combine it with the EPSON MX-80 printer and you have a system that can do about everything you would want and expect for its class.

Because of their flexibility and many features, connecting the Sorcerer and MX-80 requires a bit of thought and study. Depending on the software being used, and the kind of printing desired, changes must be made to accommodate a specific operation. These changes need not be extensive or difficult and can be considered as normal as changing the baud rate on the computer, changing the duplex setting on a modem or changing the parameters in a word processing system.

This article, though not an exhaustive one on the subject, is intended to clarify, and provide information in addition to that supplied with the unit or available from other sources.

Briefly, for those who are unfamiliar with the MX-80, it is a dot matrix printer that features a graphic character set in addition to the standard ASCII characters. Printing is done bidirectionally by a 9 wire print head, at about 80 characters per second. The ASCII characters are printed in a 7x5 dot format, except for lower case g, j, p, q and y which have a 9x5 dot format, thus providing true descenders. The expanded characters have a 9x9 dot format. The graphic characters have a 12x6 format and follow the ones found in the TRS-80 system. It should be noted that the graphic characters printed are configured in a printer ROM and do not match the characters output to a video monitor by the Sorcerer.

Printing can be done with the **standard**, **compressed** or **expanded** character options as well as **double-strike** and/or **emphasized** modes yielding a dozen different printing styles. The printer has its own vertical and horizontal tab settings, variable line spacing, as well as the form and formatting controls found on most printers. Paper widths to 9 1/2 inches are fed by a tractor drive. (EPSON recently introduced a new model printer, the MX-80FT, which has friction feed, but is similar to the MX-80 in other respects.)

A parallel interface, Centronics compatible, is standard on the printer. Connecting the MX-80 to the Sorcerer seems to be the greatest problem for those who have selected this system.

First, there is the standard 'Centronic' configuration as outlined on p. 34, Table 8 of the **Sorcerer Technical Manual** (First Edition, Mar. 1979). If your Sorcerer and MX-80 are connected in this manner and your software is written to operate with a Centronic compatible printer, then you should be printing the alpha-numeric and ASCII characters with no problem (assuming you have the switches correctly set in the printer). Try to print graphic characters and you will meet with failure. Look carefully at Table 8 in the **Technical Manual** and you see that there are only 7 data bits listed. The 8th bit of the data bus connection serves as a strobe or data available function. Since a 7 bit bus can only directly call 128 characters, and the graphics are called by characters above 128, this cable configuration is **not** suitable for some situations. What is needed is a full 8 bit data bus.

To obtain an 8 bit data bus, two cable connections must be changed to run off the parallel port. At the Sorcerer end of the cable, move the wire at PIN 4 to PIN 3. Connect PIN 4 of the Sorcerer to PIN 9 of the MX-80. You now are able to operate your MX-80 utilizing all its features. Turn on the printer, load your software into the computer and try a print run. You should now be getting those graphic characters you wanted. But wait a minute. The printer won't print any text. It's putting out graphic garbage, line feeds, form feeds, buzzers, etc. What's wrong?

Remember that the Centronics output from the computer's parallel port has the 8th bit (DATA BIT 7) operating as a strobe so that it goes high with each byte that is sent out the port. With the 8th bit always high, the printer sees all data it receives as being for characters 128 to 255. This is exactly the data for the

(continued on next page)

(MX-80 continued)

printer's graphics. To correct this situation, it becomes necessary to abandon the Centronics output and have the computer send out data via a normal parallel output method.

If your software can only operate a Centronics output, then it cannot be used with the cable wired for an 8 bit data bus. (Relax, life is not really that cruel!) You can simply use the original connections of Table 8 and accept the fact that you cannot output graphic characters with that software. Chances are that the software was written for a text situation anyway, and the unavailability of graphic characters is not a serious loss.

The 8 bit data bus cable configuration can operate with BASIC or other general purpose languages. In this situation, the Sorcerer/MX-80 team really shines. Using the output port, it can send all 256 character codes to the printer. This gives the ability to turn the printer off and on, form feed, line feed, change printing styles, print graphics and all the other functions the MX-80 is capable of doing.

To turn the printer on from a BASIC program, enter the following line in your program just before you wish to print:

```
LINE# POKE 32720,33:POKE 32721,224:PRINT CHR$(17)
```

Output to your video monitor will now cease. All PRINT commands in your program will now be sent out the parallel port to the printer. **NOTE:** since you are now printing, your program must be written to reflect that function. A screen clear for the video monitor becomes a form feed on the printer and advances printing to the next page. The video monitor provides a dynamic output from the computer while the printer provides a permanent and static record. Don't expect an animated display, for example, to successfully output to the printer.

After your program line outputting to the printer, enter the following line to turn off the printer and transfer the Sorcerer's output back to the video monitor:

```
LINE# PRINT CHR$(19):POKE 32720,27:POKE 32721,224
```

Utilizing the methods just described, I've set up a retrieval system for magazine articles on electronics. The article titles, their dates and their locations are kept in a BASIC data file containing about 250 entries. When I enter a request via the search program, the printer is turned on, the selected article file is printed, and then the printer is turned off. When the search and print routines are completed, the printer's buzzer sounds. With the search process taking from 1 to 2 minutes, I can go about doing other things and the printer's buzzer will let me know when it's done!

With the **Guided Tour of Personal Computing**, the **Sorcerer Technical Manual** and the **Epson-MX-80 User's Manual**, it should be possible for almost everyone to successfully operate their Sorcerer/MX-80 system. Get to know it, practice using it and take pride in a very versatile system.

-----

#### KEY-IN HELP - by Frank Voss

Entering a machine language program can be a tedious, tiring and time consuming task. The job can be speeded up considerably by enlisting the aid of someone to read off the hexadecimal values to you, as you key them in. This procedure significantly reduces errors. When no one is available (or willing to volunteer their assistance), I use my tape recorder.

Using the tape recorder as a dictating machine, I read off the hexadecimal codes from my hardcopy at a rate consistent with my ability to key them in. After every line is read off using the HEXPAD program published on page 26 of issue 3.2 of the **Sorcerer's Apprentice**, the job becomes quick, easy and nearly error free. As a further check, after the program is keyed in, the tape can be replayed and you can examine a memory dump for errors.

**HARDWARE NOTES - by Russell Frew, Hardware Editor**

There are numerous hardware fixes that have been released to enhance both Sorcerer I and II. Some have been published in the Sorcerer's Apprentice and some have not. I thought it might be helpful to pull together all the fixes that I am aware of and put them out in one list. The step-by-step revision can be obtained as outlined below.

There are a few cautions that must be considered. There is 110v AC inside the Sorcerer. Never work with a system that is plugged in. Also before you open your Sorcerer, make sure you understand how easily an integrated circuit can be damaged. It can be done from the static electricity in your body, a soldering iron that is not expressly designed for **digital** work, or any one of a hundred other ways. The best rule is if it works, don't mess with it! Unlike light bulbs, IC's look just the same when they are dead as they do alive. Problems can be very hard to find without considerable test gear.

**Determining Revision Level**

Open your Sorcerer slowly. The keyboard is connected to the main PC board via a 16 DIP socket. As you lift off the cover, unplug this cable while noting its orientation. With the computer sitting in front of you in its normal position, you will see a big PC board which makes up the main logic board and a smaller PC board secured above it which is the cassette interface board. The first thing to do is locate the Revision number on the main board. This is located in the left rear corner by the power supply and will be partially hidden by the cassette interface. There is a long series of hyphenated numbers and the revision level will be last in the series (eg. SS-80-9044-18-A3). Some modifications only apply to certain Revision Levels.

**Summary of Technical Revisions**

1. Sorcerer I either won't work or works very poorly with floppy disk system. An upgrade of the main PC board is required. A moderate modification.
2. Sorcerer II either won't work or works very poorly with floppy disk system. An upgrade of the PC board is required in addition to the exchange of some components. Detailed modification.
3. Micropolis systems boot at an address that conflicts with Sorcerer's ROMs. This modification moves the boot address to BC00. The fix is done on both S-100 PC board and the disk controller board. A moderate modification.
4. Cassette interface modification that sets the rest parameter for the Phase-Lock Loop Oscillator and its capture range. This fix should not be attempted by the uninitiated. It requires a 500 KHz oscilloscope, low capacitance probes and additional components. A technical modification. (See issue 3.4, p. 66)
5. A cassette/disk system that has functioned properly in the past suddenly begins to give a high number of errors. The fix is one that everyone should consider. It entails the replacement of VR1 and recalibration. Easy modification but does require resetting the VR level.
6. Failure of the shift lock to function properly. This is caused by the failure of the diodes on the keyboard. The fix is easy and, while you're right there, I would suggest the addition of the LED power-on light Howard Arrington outlined. Easy modification.
7. If you are using a board in your S-100 box that uses External Interrupt in Z-80 mode 0, you should consider this fix. In the current configuration the M1 control signal is not present and it is possible to crash your system under certain conditions. A moderate modification.
8. The TEC/CITOH daisy wheel printers have an insufficient handshake to

(continued on next page)

(HARDWARE NOTES continued)

interface with the Sorcerer. This fix is on the printer's PC board. A moderate modification.

9. If you have a Sorcerer II, Rev A3 or B, and you are considering using the EPROM Pac with 2732's or 2532's, there is a fix you must perform first. This corrects a reversed address line that goes to the Pac. Moderate modification.

10. If you are wasting a lot of time because you have a serial interface that must always be loaded and checked, consider the installation of the new Monitor 1.1 ROM. This could ease your problems considerably. The various fixes are listed in SA Issue 2.4, pg 34. Easy modification.

If one of these problems sounds like the trouble you are having, send a note to Exidy Systems Inc., 1234 Elko Dr., Sunnyvale, CA 94086. Ask Customer Service to send you a complete set of their TECHNICAL NOTES. Please note that any modification rated as moderate or better requires cutting and delicate soldering on the main logic board. It assumes a volt-ohm meter and 25 watt or less digital soldering iron as minimal equipment.

---

#### THE WORD PROCESSING CORNER - #14 - by Steven Guralnick

With this article, I will start a series on printing with SPELLBINDER. Whether you have a precision printer such as our Diablo or just a simple list device, a basic understanding of the printing options afforded the user is helpful. I'm going to start with the basic routines which the user needs to know, irrespective of the type of printer utilized. I hope I will not bore the more advanced user; however, from the number of calls and letters I get, it appears to me that the use of the simpler routines is not always as clear as it should be. Incidentally, for those of you using the PAC, you might also pay some attention to the material coming in the next few articles, because some routines are similar for the PAC and SPELLBINDER.

SPELLBINDER employs three menus for printing: two "Y-menus" and one "YT-menu". The two "Y-menus" can be switched by the command "YS". That command can be embedded in text as ".YS". Each menu programs the same functions and can be set separately, either from the console or by text. The "Y-menu" is the menu which is used for printing commands, including a selection of the printing device and all basic format instructions. The "YT-menu" is used exclusively for the setting of a title, including its location on the page, and the setting of the page number, also including its location on the page.

When you first start to run with SPELLBINDER and if you have had no previous experience with the program, you should stop, take a slow breath, and study the "Y-menu" to see what entries you should be making. The table will look like this:

```

PRINT DEVICE
PRINT ROUTINE
LINES/PAGE
CRs PAGE END
CONTINUOUS PR
INDENT-10THS
CRs PER LINE
RIGHT JUSTIFY
WIDTH-10ths
LF SIZE-48/IN
CHAR SIZE-120
SPECIAL CHAR
PROPORTIONAL
MAXIMUM SPACE
MINIMUM SPACE

```

As you can see, in our example, there are no numbers following the entries. In forthcoming articles I will describe how to set the entries.

### TRS-80 PROGRAM CONVERSION - by Larry Stempnik, Technical Director

There is a wealth of TRS-80 programs available on **Download 80** systems via modem. The first problem to attack in translating these TRS-80 programs, is the software conversion using ASCII files. A quick summary of the procedure is:

1. Download the TRS-80 program with a modem from a network or a friend with a TRS-80, level II, expansion interface, and a modem.
2. Download from the Quality Software Smart Terminal file into the Word Processor Pac. Print the file double spaced with 25 lines per page. Save Word Processor file on tape and shutdown.
3. Quickly scan the printout to determine the amount of effort required to convert and whether it is worth it.
4. Mark up the printout to shorten lines to less than 64 characters. A little effort here results in a much better display of long descriptive paragraphs of text. Add any logic changes or program modifications. This is better done off-line since you can quickly scan the printout for conflicts or logic branching. Do not bother with routine substitutions here.
5. Plug in the BASIC Rom Pac and turn on the power. Load in the relocated Word Processor tape and go to the Command mode. Load in the tape saved in step 2.
6. Do all the automatic changes using the 200s/old/\*new/ command. As Steve Guralnick says, use the \*, you never know when it might save you from creating errors. Do your replacements as follows:

```
200s/'/* REM /
200s/?/* PRINT /
200s/CLS/* PRINT CHR$(12) /
200s/PRINT CHR$(26)/*PRINT /
200s/OPEN/* REM OPEN/
200s/CLOSE/* REM CLOSE/
```

These commands will perform all of the simple replacements that are required.

7. Scan for ELSE commands at the end of IF-THEN statements. Move the cursor to the first E in ELSE, hit the expand key and type a return. The first line is now a standard IF-THEN line. While still in the expand mode, type a line number and the IF with the condition reversed. Then close up the expand mode, rubout the ELSE, add THEN, and leave the rest. The result is now Sorcerer compatible. For example:

```
500 IF A=B THEN PRINT "EQUAL" ELSE PRINT "UNEQUAL"
```

This would become:

```
500 IF A=B THEN PRINT "EQUAL"
502 IF A<>B THEN PRINT "UNEQUAL"
```

Search for "@" and USING with the search command. The "@" usually occurs in PRINT@ statement and must be converted or the Sorcerer will wipe out the whole line. Refer to the following discussion on how to handle these conversions.

8. Search the text on the screen from top to bottom for any other changes required such as shorter lines. The best way to correct long lines is to position the cursor over the start of a word near the end of the line, hit the expand key and type a return, new line number and the appropriate start of the next line, such as PRINT, then close up the text. Another method is to assign a string variable name, such as A\$, to the text on one line and print the text on the next line with a PRINT A\$ statement.

(continued on next page)

(TRS-80 PROGRAM CONVERSION continued)

9. Save the corrected text on tape as a Word Processor tape in case future changes are needed or desired. Next, enter the Command mode and type J to go to BASIC. (See the Sorcerer's Apprentice Issues 3.1, pg 4 and 3.3 pg 56 for the J command). Watch the screen for error messages as the program transfers. It can be slowed down by using SET S=10. Run the program and note any error messages. If the errors are not obvious, print a listing of the BASIC program, and compare it to the WP printout. It is not necessary to CSAVE a BASIC tape at this point since the WP tape can be used just as easily with the J command.

10. The above procedure was all that was required for program conversion of non-graphics programs. If your program is not running at this point, then try to understand how the original program works and test the converted program section by section until you find the problem.

The following commands are equal and no modifications are required.

**Mathematical functions:**

SGN, INT, ABS, SQR, RND, LOG, EXP, COS, SIN, TAN, ATN

**Operators:**

+, -, \*, /, ^, AND, OR, >, =, <

**Commands:**

RUN, LIST, CLEAR, CLOAD, CSAVE, NEW

**Expressions:**

END, FOR, NEXT, DATA, INPUT, DIM, READ, LET, GOTO, IF, GOSUB, REM, STOP, OUT, ON, POKE, PRINT, TAB(, TO, THEN, NOT, STEP, PEEK, LEN, STR\$, VAL, ASC, CHR\$, LEFT\$, RIGHT\$, MID\$, RESTORE,USR, INP

**TRS-80 disk commands NOT used (in a tape system):**

CMD, LINE, OPEN, FIELD, GET, PUT, CLOSE, LOAD, MERGE, NAME, KILL, LSET, RSET, SAVE, LPRINT, DEF, FN, INSTR, TIME\$, CVI, CVS, CVD, EOF, LOC, LOF, MKI\$, MKS\$, MKD\$

**ARRINGTON SOFTWARE SERVICE**

**UPGRADES AVAILABLE** - In the April 15th issue of this Newsletter, we advertised that Galaxians was available with sound. However, due to a belated bug in the sound and joystick routines, we continued to ship the previous non-sound version until May 26th. If you ordered Galaxians between April 15th and May 26th, and would like to have the sound version as advertised, please send a letter asking for your FREE update. All others who would like to upgrade to enjoy fantastic sound in both Galaxians and Invaders, may do so by sending \$4.00 to upgrade either program, or \$6.00 to upgrade both at the same time.

I'm sorry about the frustration some of you felt when you tried to implement our sound article for Invaders published in the April 15th issue. Although the code works as published, those of you who bought Invaders from System Software found that you could not save an upgraded version because they have copy protected their product. We invite you to make your future purchases from Arrington Software Service and avoid this kind of inconvenience. The sound in our latest version of Invaders is FANTASTIC, and is not to be confused with the trivial sound routine published in the article.

The following catalog includes 5 new tunes for the music system. Tunes are now marketed 2 per tape for \$10.00. If you have already purchased one of the two tunes on a tape, such as Boogy, you may buy the companion tune (i.e. Eleanor) for \$5.00.

## ARRINGTON SOFTWARE SERVICE

### MUSIC SYSTEM

①

**FEATURES** -- Tempo control from slow to exceedingly fast.  
 Set the music's key signature by placing sharps or flats on the staves.  
 Select notes by moving cursor up and down musical staves. It's like copying what you see on music sheets.  
 Graphically select note durations and rests.  
 Full editing of inserting, deleting or overwriting chord sets in the song data file.  
 Hear a pitch in a chord before the note is entered into the data file.  
 Play the last ten chord sets to verify the accuracy of the data just entered. Or play the entire song.  
 Copy refrains for rapid duplication of repeated measures.  
 Transpose the playing of the song into another key. Adjust to suit your vocal range for singing along.  
 Automatic loading and saving of data files from and to cassette.  
 Restoration of data file notes to the screen for rapid editing and file verification.  
 Single step through the file listening to each chord. Aids in file editing.  
 User prompts and error protections. System includes the necessary hardware interface to your amplifier.  
 Select individual voices to play only their part. Example: hear the Bass and Soprano parts together.

**EASY TO USE** -- Customers, DON'T BE HESITANT because you feel you know nothing about music. You don't need to know all about music to enjoy this software. You can experiment with the editor, dabble in song writing, and enjoy prewritten music such as PIANO PLAYER, FANFARE, BACH, FINLANDIA and BOOGIE WOOGIE. Creating song data files is done graphically.

### PIANO PLAYER

This program is option for the above music system and adds delightful graphical animation of a high resolution piano player tinkling the ivories. The little man's arms move in synchronization with the beat of the music. A large keyboard is displayed upon which four cursors jump around on the keys to the four notes being played. Piano player comes with another sample song file. Being both graphical and musical, this program is a crowd pleaser twice over.

②

### MACHINE CODE TUTORIAL

Many have requested additional knowledge about your Sorcerer and help in learning to program in machine language. This package will help you understand video and keyboard routines, Z80 registers, instructions and flags, Monitor and Basic work areas and useful routines in ROM, and sound generation. I particularly like the interactive environment where one can immediately try the exercise ideas being presented. The eight programs lead the user step-by-step in their development. Highly recommended! (Even though I consider myself well versed, the author taught me some new tricks. HA)

### DATABASE SYSTEM II

The program is written in machine language and stands alone as a complete database manager. Features include storing up to 500 records consisting of up to 9 fields. One can sort on any field, or search any field for either a partial or complete match. The powerful report feature allows one to arrange the placement of data fields for printing of mailing labels, for example. The RANGE options allow you to select one, ALL, or those records that satisfy the field match criteria. The database is cassette based, Files can be loaded, saved and merged. Edit features include ADD, DELETE, EDIT. One can elect to use their own printer driver, the serial driver, or the parallel printer driver. The latest feature to have been added is a Column TOTAL with a range of +-999999999.99.

### GRAPHICS PACKAGE II

Machine language routines manage all 128 Graphic characters to give the programmer ultimate flexibility in plotting in high resolution of 512x240. You pass to the routine the coordinates of the lines or points to plot, and the routines do the rest with graphic characters until all 128 are used. As characters are freed up, they rejoin the pool of available characters, all of which is transparent to the user. If an existing character pattern matches the one needed, it is reused rather than defining a redundant character cell. The mileage one can get out of 128 characters is amazing. Coordinates are passed in variables for plotting or erasing both points and lines. Documentation is thorough and one can use the routines with the ROMPAC Basic or other languages that can call machine code routines.

③

## SORCERER INVADERS

SORCERER INVADERS is a fast action game similar to the popular arcade version. The rows of marching invaders continue their advance while you dodge back and forth avoiding the falling phasers. The bunkers under which you may hide only provide temporary protection. The only hope for you is skill in shooting down the advancing party, each and every one of them. Then to your horror another screen full of invaders appear to continue their march back and forth and downward. The superb graphics and the fast-paced action make the excitement very real. The game is addictive because it is so fun. Invaders comes with both joystick or keyboard control, and sound.

**JOYSTICK CONTROL:** Joystick control is via the parallel port and follows the standard adopted by Global Software Network wherein Left is input on pin 10, Right is input on pin 22, and Fire is input on both pins 10 and 22.

**KEYBOARD CONTROL:** The player may also use the keyboard to control the movement of his missile pod by pressing the '<' and '>' keys. The TAB SKIP is used to fire back at the Invaders.

**SOUND:** Sound is output to the parallel port bits. For safety's sake, we recommend that you use the board from our MUSIC SYSTEM to interface the parallel port to your amplifier and speaker. The sound is excellent and adds to the drama of the Invaders marching closer and closer to landing. The beat gets faster and faster just like in the arcade version! The game's sounds make this version superior to any a competitor may be marketing for the Sorcerer.

## GALAXIANS

SORCERER GALAXIANS is a fast action game similar to the popular arcade version. Galaxian spaceships peel out of formation at the top of the screen, and fire at you as they dive and zip across the screen. You constantly dodge back and forth at the bottom of the screen trying to shoot down the darting ships while avoiding them and their fire. The superb graphics and the fast-paced action make the excitement very real. The game is addictive because it is so fun. Galaxians comes with both joystick or keyboard control, and superb sound!

**JOYSTICK CONTROL:** Joystick control is via the parallel port and follows the standard adopted by Global Software Network wherein Left is input on pin 10, Right is input on pin 22, and Fire is input on both pins 10 and 22.

**KEYBOARD CONTROL:** The player may also use the keyboard to control the movement of his missile pod and its firing by pressing the left and right arrow keys on the 10-key pad. The space bar is used to fire back at the Galaxian ships.

**SOUND:** Sound is output to the parallel port bits. For safety's sake, we recommend that you use the board from our MUSIC SYSTEM to interface the parallel port to your amplifier and speaker. The sound is great, and doubles the enjoyment and interest of the game, although it does slow the game down somewhat. If you prefer to have the maximum speed of the game, you may eliminate the sound by simply selecting a different )GO address. However, I believe most would rather not give up the spaced-out sounds that accompany this version from Arrington Software Service. Don't miss out on the best by buying GALAXIANS from any other vendor.

## DISASSEMBLER

This program is a Z-80 machine language two-pass disassembler whose output format is directly compatible with the Sorcerer Development Pac. The Z-80 assembly language source (input to assembler) listing can be sent to Video, Cassette or Printer. The cassette file produced is a source file for the Editor/Assembler and can be read directly into the Editor or Assembler of the Development Pac. The disassembler has a displacement function which allows any program residing anywhere in memory to be decoded, whether it is at it's normal address or has been moved to be decoded.

SAMPLE PRINTOUT	E993	F5	LE993	PUSH	AF
	E994	CD 1B E0		CALL	LE01B
	E997	FE 0A		CP	00AH
	E999	29 14		JR	Z,LE9AF-4
	E99B	F5		PUSH	AF
	E99C	DB FF	LE99C	IN	A,(0FFH)

**SORCERER'S APPRENTICE, VOL 3, NO. 5**  
**ARRINGTON SOFTWARE SERVICE**  
 9522 LINSTOCK, BOISE, IDAHO 83704

FINLANDIA & FANTASIA - Renditions of Jean Sibelius's FINLANDIA and Wm. Byrd's FANTASIA. Music system song files.  
 BACH - Exceptional rendition of Bach's "Jesu, Joy Of Man's Desiring" and "ODE TO JOY". Music system song files.  
 BOOGIE & ELEANOR RIGBY - Very lively with walking bass parts. Really has a dance rhythm and lots of bass notes.  
 THE STING & MAPLE LEAF RAG - Scott Joplin's all-time favorites are vividly alive and exceptionally well done for music.  
 MUSIC SYSTEM - Our finest piece of software with 4 part harmony, hardware and exceptional editor using graphics.  
 PIANO PLAYER - Graphical animation of a piano player playing your Music System songs. 4 cursors bounce on keys played.  
 OTHELLO - Challenging machine language algorithm and speed make this fun to play against computer or friend.  
 DISASSEMBLER - Creates machine language source code on CRT, printer or tape which is fully DEVELOPMENT PAC compatible.  
 QUBIC - My best artificial intelligence program. Practically unbeatable as you try to get 4 in a row on 3-D grid.  
 QUICK EDIT - Simple screen editing for rapid creation of graphical displays, text and data files.  
 EDITOR FOR BASIC - Complete editor, easy to use, has revive, renumber, insert, delete, and tab features.  
 SUPER-X EDITOR - Similar to EDITOR FOR BASIC, with additional features: string search, block delete & list, merge, compact program, block renumbering, RUN/STOP listing pause, execute Monitor commands from Basic mode.  
 CROSS REFERENCE - Complete listing of variables and the line numbers where used. Also references all GOTO, GOSUB, etc.  
 GRAPHICS PACKAGE I - Resolution of 128x90. Fantastic demonstration includes screen motion, X-Y-Z plotting, bar charts, circles, and stars. Figures can occupy the entire screen if desired. Fast and easy to access routines.  
 GRAPHICS PACKAGE II - Resolution of 512x240 using X & Y coordinates. Includes superb demonstration. Figures are limited to 128 unique graphic cells. Both Graphics I and II include point and line drawing features in machine code.  
 ARTILLERY - Requires GRAPHICS PACKAGE I. Players fire cannon balls at each others castles until one lands a direct hit.  
 SPACE INVADERS - Sorcerer version of the popular arcade game. Fast paced. EXCELLENT graphics. Very addictive game.  
 MUSICAL HORSERACE - Animated horses race while one-voice music plays. Quite an attractive betting program.  
 CONCENTRATION - Mentally challenging match game utilizing graphical figures. Up to 4 players or against the computer.  
 JAIL BREAKOUT - Keep the balls bouncing against the brick wall until all bricks are removed. Similar to TV game.  
 CHESS 'BRUCE' - At last, an original chess program for the Sorcerer. It's GOOD, but hasn't yet competed against others.  
 CASSETTE FILES - Open, close, read and write strings to a buffer. Buffer automatically loads from or writes to tape.  
 BLACKJACK - Graphical cards, easy user inputs, and real Las Vegas style. 1 to 5 players. Very nice game.  
 SPACETREK - Excellently controlled screen display of all scanners and status. Has sound, some graphics. Well written.  
 DATABASE SYSTEM - Add, Create, Delete, Edit, List, Load, Merge, Printer, Query, Report, Sort, Tabs, Write, Column TOTAL.  
 SCREEN GENI - Utility for inverse video printing. Enhancements give effect of print at, and print using control.  
 GALAXIANS - Truly the rival of all arcade games! Ships peel out of formation and zip across the screen firing at you.  
 MACHINE CODE TUTORIAL - 8 taped lessons teach machine language, Basic, Monitor, RomPac routines. Interactive exercises.  
 JOYSTICKS - Atari joystick pair modified to connect to Sorcerer's parallel port. Includes demonstration software.

Your order will be in the return mail within 3 days. Software is recorded at both 300 and 1200 baud. I guarantee my software and have a strong desire to have customer satisfaction and am willing to try to answer any questions.

HOWARD ARRINGTON  
 9522 LINSTOCK  
 BOISE, IDAHO 83704

(208) 377-1938 After 6 p.m.

Mountain Standard Time.

Checks O.K. - No credit cards.

[ ] MUSIC SYSTEM	\$40.00	BMUSHE	SORCERER SIZE>>>	16K	32K	48K
[ ] PIANO PLAYER	\$15.00	MF	RETURN ADDRESS:			
[ ] FINLANDIA FANTASIA	\$10.00	F				
[ ] BOOGIE & ELEANOR	\$10.00	F				
[ ] BACH'S GREATEST	\$10.00	F				
[ ] STING & MAPLE RAG	\$10.00	F	[ ] ARTILLERY	\$10.00	BMG	
[ ] GALAXIANS	\$19.95	MGSJ	[ ] JOYSTICK PAIR	\$39.95	BMUH (+\$5 OVERSEAS POST)	
[ ] SPACE INVADERS	\$17.95	BMGSJ	[ ] SUPER-X EDITOR	\$21.95	MU	
[ ] CHESS 'BRUCE'	\$17.95	MG	[ ] DATABASE SYS II	\$29.95	MU	
[ ] GRAPHICS PACK I	\$25.95	BMU	[ ] M.CODE TUTORIAL	\$25.95	ME	
[ ] GRAPHICS PACK II	\$25.95	BMU	[ ] SCREEN GENIE	\$14.95	BMU	
[ ] DISASSEMBLER	\$17.95	MU	[ ] CASSETTE FILES	\$14.95	MU	
[ ] CROSS REFERENCE	\$14.95	MU	[ ] SPACETREK 32K	\$14.95	BGS	
[ ] MUSICAL HORSERACE	\$10.00	BMGS	[ ] BLACKJACK	\$10.00	BG	
[ ] JAIL BREAKOUT	\$10.00	MGS	[ ] QUBIC	\$10.00	BG	
[ ] EDITOR FOR BASIC	\$10.00	MU	[ ] OTHELLO	\$10.00	MG	
[ ] QUICK EDIT	\$10.00	MU	[ ] CONCENTRATION	\$10.00	BG	

KEY: B-Basic M-Machine code U-Utility G-Game S-Sound J-Joystick or keyboard H-Hardware E-Education F-Music File

## PASCAL PORT - by Daniel Conde

One of the problems that micro computer owners encounter is the lack of available RAM space. This is quite evident to Sorcerer owners who are bound to an upper limit of 32K or 48K without special facilities, such as the Expansion Box.

These problems arise when we want to allocate large data structures, such as a multi-dimensional matrix. We certainly cannot allocate a 1000 by 1000 matrix, but often, we want to access a limited number of items, such as 1000, in a convenient array index convention, ( array[x,y] ). Wise use of data structures will allow you to retain the index convention, and avoid having to allocate huge amounts of memory. The data structures that will help you solve the mentioned matrix problem otherwise known as a sparse matrix, could be applied to other types of problems as well.

In Pascal, the TYPE RECORD will enable you to have more abstract structures. They will allow you to bind together several different types of objects in one convenient package. To illustrate, let us define a TYPE PERSON as follows:

```
TYPE PERSON = RECORD
    NAME      : STRING;
    BIRTHDAY  : STRING;
    NUMBER    : INTEGER;
END;
```

The TYPE STRING could be an ARRAY OF CHARS, or a system supplied TYPE, depending on the installation.

Now, returning to the sparse matrix problem, let us define another data TYPE to act as **cells** in a matrix.

```
INFO = RECORD
    DATA      : PERSON;
    XCORD      : INTEGER;
    YCORD      : INTEGER;
END;
```

Finally, since we know that we are going to use only 1000 of these cells in the proposed 1000 by 1000 matrix, let us **simulate** that by making another declaration:

```
MATRIX = ARRAY [ 1 .. 1000 ] OF INFO ;
```

In the program, we could have a variable BOOK of type MATRIX, and we could access individual fields by the standard record convention, such as BOOK[233].DATA.NAME for the name field, or if you want to check the coordinates only, we may have a statement:

```
IF(BOOK [ INDEX ],XCORD = DESIRED_X_COORDINATE) AND
   (BOOK [ INDEX ],YCORD = DESIRED_Y_COORDINATE)
   THEN PRINTOUT( BOOK [ INDEX ].DATA ) ; (*<-your procedure!*)
```

In our data structure, we will actually need to check the x and y coordinate of each of the cells in the simulating array to access a desired cell indexed by [x,y]. It is tedious, but then, you will save space. Had the array been sorted by the x and y coordinates acting as keys, the search time could be decreased.

Although having many TYPES and SUBTYPES may seem troublesome, they actually allow you to detach the programmer from the nitty-gritty details of how the data is implemented. Also, modifications may be made to lower level declarations, and the program may need **minimum** modifications on the higher levels.

I hope I've shown you how to implement cleaner data structures as well as how to simulate a large array in a limited address space.

## DISK NOTES - by Bryan Lewis, CP/M Editor

Ahh.... I got my first "Letter to the Editor". I received a letter from Roger Hagan that asked, "Using CP/M, how may I load a machine language utility program to a specific address from disk (as I can from tape), without overwriting an application program already in memory?" The answer, very briefly, is: there isn't any ready-made way to do it. It can be done, but I don't think anyone has reduced it to practice yet. I'd like to discuss why and then outline a proposed solution.

We Sorcerer users are accustomed to being able to load programs from cassette at any address. The proper address is saved right along with the program in the tape header, or we can even specify a different address in the LO command. Some disk operating systems do the same. Micropolis' MDOS, for example, has a LOAD command with very similar conventions. But CP/M doesn't have an intrinsic (i.e., built in and always in memory) LOAD command. All COM files are loaded at 100 hex, into the Transient Program Area (TPA).

There are ways to put files at other locations, of course. We can use the DDT utility: either read the file into the TPA, then move it to where we want it, or else read it directly to the desired location by specifying an offset in the read command. Then DDT's G command will start execution.

That's too much work, though. We would prefer to save the address information with the file and not have to remember the numbers each time we reload it. One way to do that, the brute force method, is to save all of memory up to the end of the file. We only have to add a jump instruction at 100H to transfer execution to the proper address. For an illustration, let's say we have a program that occupies 1000H to 17FFH:

```
Cold boot CP/M.
A>DDT          DDT is one way to exit to the Monitor.
-GE000
>LO           Load the file from tape (at 1000H).
>EN 100      Enter the jump instruction.
0100: C3 00 10 / Means: Jump to 1000H.
>GO 0        Warm boot CP/M.
A>SAVE 23 NAME.COM From 0100H to 17FFH is 23 pages.
```

That will waste a lot of disk space, however. We can do better by moving the file down to 110H and adding a small relocator routine when we save it. The relocator will sit at 100H and hence will be the first thing executed when we run the file later. After the relocator we'll put the same jump instruction we mentioned above.

```
Cold boot, exit, and load as above.
>MO 1000 17FF 110 Move it down.
>EN 100      Manually enter the relocator:
0100: 21 10 01    LD HL,0110H    ;From.
0103: 11 00 10    LD DE,1000H    ;To.
0106: 01 FF 07    LD BC,07FFH    ;Number of bytes.
0109: ED B0      LDIR          ;Move it all.
010B: C3 00 10 / JP 1000H      ;And go to it.
>GO 0        Warm boot.
A>SAVE 9 NAME.COM Only 9 pages now: 0100H to 090FH.
```

For any given file, you'll have to change the destination address (at 0104H), the length (at 0107H), and the execution address (at 010CH). Caution: I've used the LDIR instruction here, which means that the move is made from the bottom up. If the file had been bigger, so that the moved-to area overlapped the moved-from one, that would result in some of the file being overwritten before it had a chance to be copied. The LDDR instruction (top down) would then be better. Sketch it on a piece of paper before you do it.

(continued on next page)

(DISKNOTES continued)

This method is essentially the same as the ones published in earlier newsletters and on the Sorcerer Bulletin Board, for saving RomPac programs on Micropolis disks. (A relocater is needed even with MDOS; its LOAD command won't overlay files into the same area of low memory occupied by the operating system.) The DDT utility in CP/M is an example of a self-relocating program, by the way.

We still haven't solved the problem of wiping out any previously existing program. All of the above methods start by loading **something** into the TPA. Can't we make CP/M load directly to an arbitrary address? Yes. The CP/M manuals call that address the DMA, which is a poor name since it gets confused with Direct Memory Access, a hardware-related buzzphrase. Think of it as the Disk-to-Memory Address. It's simply the location in memory that the next record (128 bytes) will be loaded at. We can specify it by making use of the handy BDOS functions provided by CP/M. Function number 26 is "Set DMA Address". We call it by putting that function number into the C register, putting the address into DE, and calling BDOS at its entry point of 0005H. For example:

```

0E 1A          LD    C,26          ;Get ready to use SETDMA ftn.
11 00 10      LD    DE,1000H      ;Specify DMA at 1000H.
CD 05 00      CALL 5             ;Call BDOS.

```

Then when we read the next 128-byte record from the disk (by calling the READ function in BDOS), it will be loaded at 1000H. For each succeeding record, we add 128 and re-set the DMA. We continue in this fashion until we reach the end of the file, which we'll know by a value passed back to us in the A register after a READ function call.

Notice that we're actually writing our own loader. That's not as hard as it sounds, since all of the disk operations are handled as canned functions we can call by number. But don't expect to get much other work done on the day you sit down to do it.

If you would like to learn more about using the BDOS functions, I recommend the CP/M Connection series of articles in the magazine **Microsystems**. The May/June issue presents and explains a print utility which uses all of the common functions.

Wait a minute -- even if we're willing to write that more flexible loader, where will we put it? If it's just another COM file that gets read from disk into the TPA, we haven't gained a thing. Therefore the loader has to be in memory all the time, somewhere out of the way of other programs. We might try squeezing it into the space left over in the BIOS area of CP/M, but I doubt that it will fit into the two to three hundred bytes available. We're better off making it a separate section of code, situated just below CP/M in high memory, or above it (by reconfiguring CP/M for a 1K smaller system).

It would be nice if we could include the proper loading address as part of the file when we save it. A simple way to do that might be to put the address into the file's name. All files with a nonstandard load address would then have names like NAME1000.COM or just NAME1000, or maybe NAME.100 if the fourth digit will always be zero. The new loader utility can easily be made to pick the address out of the name.

One final detail: how to execute the loader when we want it. CP/M doesn't have an intrinsic GO command. So let's pick a control character not needed for anything else (say CTRL-G as a mnemonic for Get). We can modify the keyboard input section of the BIOS to check for that trigger character. Schematically, like this:

```

CONIN:      CALL KEYBRD      ;Get a key.
            CPI    7         ;Is it a CTRL-G (ASCII 7)?
            JZ    LOADER     ;If so, go to it.

```

(continued on next page)

(DISKNOTES continued)

Don't be confused by the switch to 8080 mnemonics here; they're the required form for the CP/M assembler. (Sorcerer users tend to use Z80 mnemonics; I still think Z80 whenever I'm doing a quickie hand-assembled patch.)

Since most Sorcerer CP/M users don't have the source code to their BIOS, here's how to find the CONIN section. (I don't have room this month to discuss how to make the patch and get it onto the disk. Experiment.) First look at location 0 in memory; it contains a jump to the warm boot entry point. For a 32K system, you might find:

0000: C3 03 75

The CONIN entry is 6 bytes past that (as explained in the CP/M System Alteration Guide). So look at 7503 + 6:

7509: C3 E0 7C or whatever it is.

Finally, at the end of that jump you should find some code that resembles a keyboard input routine, like:

7CE0: CD 09 E0 ....

---

### DIABLO HYTYPE SERIAL DRIVER

This routine operates under MDOS at 1200 baud. Boot MDOS and patch the following into the RES module:

```

050A:CB 06 ;Where to jump for PRINTing
06CB:E5 PRINT PUSH HL ;Save registers
06CC:3E C0 LD A,C0 ;Tell port baud rate
06CE:D3 FE OUT (FE),A ; C0 = 1200 baud
06D0:CD E4 06 CALL DELAY1 ;Go wait
06D3:78 LD A,B ;Get char from MDOS
06D4:CD 12 E0 CALL OUTAPE ;Monitor TAPE OUT
06D7:CD E4 06 CALL DELAY1 ;Wait again
06D4:DB FD READY IN A,(FD) ;Check if port ready
06DC:E6 01 AND 01 ;
06DE:CA DA 06 JP Z,READY ;No? Loop
06E1:E1 POP HL ;Restore registers
06E2:AF XOR A ;Clear flags
06E3:C9 RET ;Done, go get more
06E4:21 00 05 DELAY1 LD HL,0500 ;Put loop count in H
06E7:2B DELAY2 DEC HL ; end point in L
06E8:7C LD A,H ;Get count
06E9:B5 OR L ;Compare end point
06EA:C2 E7 06 JP NZ,DELAY2 ; Done? Nope, stay!
06ED:C9 RET ;Yup! Go home!

```

If you find the routine outruns your printer, increase the delay value at address 06E6 from 05 to 06 or 07. Decrease the value if you think your printer could run faster.

You will also need to enter some nulls (NULL X) to allow time for carriage returns. Eighty column lines will require 15 nulls, while 135 column lines will need 20 nulls.

---

 \$2/line

&lt;&lt;&lt;&lt; CLASSIFIED ADS &gt;&gt;&gt;&gt;

\$2/line

---

**FOR SALE: Exidy Sorcerer** - 32K RAM (Includes all hardware schematics) some software, two cassette recorders and interfacing. Video 100 (12") monitor. Basic RomPac and Eproms for Assembler Pac. \$850. Contact Al Schmidt, Rt. #3, Box 206, Oswego, Ill 60543. PHONE: (312) 554-3044.

**EVALUATIONS** - by Emiliano C. De Laurentiis, Product Evaluations Editor

This is the first of a new column for the Sorcerer's Apprentice dedicated to professional evaluations of Sorcerer hardware and software products. The purpose of the column is to provide objective evaluations of these products so that potential users may judiciously decide whether or not a particular product is worth purchasing and using.

I will evaluate any product for the Sorcerer submitted for evaluation. I would prefer if the vendors themselves contribute their products for evaluation, since that would indicate the dedication of the vendor to the support and confidence of the product.

My current interests are to evaluate cassette-based business software, utility software, and entertainment packages. These are intended as reviews of the final product.

With this first column, I will review **Graphics Package II** (originally released as **Parsiplot**), a high resolution (512x240 dot) plotting utility written by Geoffrey Cockhill and distributed by Arrington Software Service. It makes use of the 128 programmable graphics to plot on the Sorcerer screen.

Since it is well known that the Sorcerer's graphics capabilities is limited to programmable characters, rather than to having the more flexible bit-mapped screens, it is not necessary to emphasize the importance of being frugal with how one uses programmed characters. (All Sorcerer owners should write Exidy lobbying for them to include a bit-mapped display in the Sorcerer III.) In view of this limitation, **Graphics Package II** will never define a graphic character which is a repetition of a previous character. What this means, then, is that one may obtain quite elaborate plots in 512 x 240 resolution!

The use of this program is simplicity itself. It is delivered for use with the Basic ROM PAC but may be called from other languages. In Basic, a USR call will allow turning individual points on or off, testing of the existence of a point, or drawing and erasing of lines. For the latter, it is only necessary to give the two endpoints, and the line is drawn.

The documentation is well written and should not present any problems, no matter what your level of programming expertise. Furthermore, information is provided which would help in interfacing **Graphics Package II** to other languages. There are also a number of special features mentioned that help in the development of software. For example, a counter is available that indicates how many graphic characters have been used up. A full listing of the demonstration program is also provided. This should be much appreciated by those who do not own printers since it allows easy study of the use of the routine when developing software.

In terms of design, **Graphics Package II** is a very elementary piece of software. Think of it...all it can do is draw points and straight lines. As a product, It is well written and is another good example of fine planning. There are a number of improvements to be made to the software, though. It would have been a much improved product if it had the following features:

- 1) A screen save feature so that plots could be recalled immediately from memory.
- 2) A plot-to-alternate-pages feature, so that the system draws to alternate pages while the user is viewing the current page.
- 3) For each page, the unique set of graphic characters should be saved along with the page, so that one would not run out of graphic characters.
- 4) A flip page routine, where alternate pages are moved to the screen to form some animation.
- 5) Draw circle and polygon routines.
- 6) Rotate objects, and expand or shrink objects routines (although these may be easily written in Basic).
- 7) Move object routines.

In the next issue: a review of **SORCERER ASTEROIDS** and **SKETCH**

**DUSTINGS FROM THE LIBRARY** - by Robert Hageman, Librarian and SysOp

In this column I'll give you more tips on how to streamline your use of the on-line system.

Usage of all the '?'s with the DIR and TYPE commands, as discussed in the last issue, can be replaced by '\*', e.g. for all L files type DIR L\*.\* . Use '?' as space holders, e.g. to list all three letter name files with ???.\* .

**Control H** does **not** work at RCPM system level. Use of control H is limited to only certain programs, e.g. MINICBBS and CHAT.

When you are at the RCPM system level and make corrections in a command line, use **control R** to have a clean line retyped. This eliminates the echoed erasures and just leaves the characters the system will interpret. For example, if you make an error in TYPEing out the PRINTER1.TIP file, a control R will retype the line, minus echoed erasures:

```
A>TYPE PRNTER11RETNINTER1.TIP# (control R prints the #)
TYPE PRINTER1.TIP (system response to the control R)
```

When you TYPE a file, the '.' is necessary between the file name and file type. Forgetting to use the period will result in a "FILE NOT FOUND" error message.

**Control C** can be used in several places on the system. It will bypass the system sign-on after you have entered the number of nulls needed by your system. After you have gotten beyond the next message, "ENTER PASSWORD:" and are in MINICBBS, control C will bypass the welcome. Another control C will now bypass the bulletins. (I don't recommend the second control C since the bulletins will be used for system announcements).

Here are the file types now used in our RCPM files and an explanation of their meaning:

- 1) ADS: advertisements from this newsletter.
- 2) AHL: experimental....automatic hex loader, for use with moved and modified word processor (see columns by Larry Stempnik in previous issues).
- 3) ASM: assembly language source code in 8080 mnemonics.
- 4) BAS: Basic language program (some include tips) in ASCII.
- 5) CCM: system level commands (cannot be TYPed or downloaded).
- 6) CPM: high quality documentation of CP/M especially useful to assembly language programmers.
- 7) DOC: documentation.
- 8) HOW: also documentation.
- 9) INF: information.
- 10) OBJ: CP/M transferable CCM files (not available as commands on RCPM).
- 11) RCP: recipe. That's right there is one on the system (a great cheesecake recipe).
- 12) REV: review.
- 13) TIP: helpful hint(s).
- 14) Z80: assembly language source code in Z80 mnemonics.

If anyone wishes to use other file types on the system, please leave a brief explanation on the MINICBBS.

Here are fast means to key input for MINICBBS and message searches:

- a. Command strings may be built up in MINICBBS as a method of answering questions before they are asked by the program. When CBBS asks for your first name, enter your first name followed by ';' and your last name, e.g. BOB;HAGEMAN. This answers both **name** questions and the program will proceed to log-in.

(continued on next page)

(DUSTINGS continued)

Function requests may be answered by: **S;1**, **Q;5** or **G;N**. These tell CBBS the function requested and answers the next question (i.e. what number message or comments Y/N). For multiple message retrieval, for example, use: **R;1;5;23;45**.

- b. To do a selective search in the **S** (summary) function of MINICBBS or CBBS, reply: **S;1,T=MIKE**, **S;1,F=MIKE** or **S;1,S=MIKE** when it asks for the function. (**T** means **TO**, **F** means **FROM**, **S** means **SUBJECT** and **MIKE** is any string).

The first command means: **SUMMARIZE**, starting with **MSG #1**, for the **TO** field containing **MIKE**. You can start the selective summary with any message number, but it's so fast that I usually start with 1.

More examples: **S;1,T=KEITH**, **S;1,F=JEFF**, **S;1,S=MODEM** would print all messages to **KEITH**, all messages from **JEFF**, all messages with the subject of **MODEM**. Remember, this works only in the **SUMMARY** function. (Thanks to Keith Petersen, for providing this selective search.)

MINICBBS has an often overlooked **EDIT** feature which can be used after you've entered your message. The best thing you can do **before** you save (**S**) your message is list all of it (**L;1**). Then, if you have any glaring errors (incorrect phone number, address, etc.), you can edit (**E;n**) the incorrect line (**n**).

Besides the current CP/M files found on the **B** drive, we are assembling other public domain programs which will include Basic and machine language software. We expect to begin rotating these disks in that unit shortly. When we are ready to begin this new service, a time table will be posted to the Bulletins in CBBS and in this column.

Other RCPM-CBBS projects underway include: a new program for non-CPM users (to give the capabilities of sending and receiving CPM files to those without a disk system); and a method of down-loading machine language programs to the Smart Terminal/relocated word processor. A search is still underway to find an acceptable up-load program to allow users of Smart Terminal to transfer files to our system. We are also looking for a program capable of taking files stored in compressed Basic mode and listing them to the console. This would allow 25-30% more Basic programs to be stored in the same space occupied by equivalent ASCII files.

Glenn Forter of Download-80 advises me that as of mid-April, Download-80 is a subscription service, and requires a password (individually assigned). To subscribe contact Glenn via Forum-80 of Mount Clemens, MI. The correct modem format for 80 systems is even parity with 7 data bits and 1 stop bit at 300 baud.

-----  
\$2/line

<<<< CLASSIFIED ADS >>>>

\$2/line  
-----

**FOR SALE: Astronomy Cassette Programs** for Sorcerer: e.g. **Planet Finder** displays RA and DEC for selected planet and then displays zodiac star chart region (each 4 RA by 60 DEC) and shows planet in constellations for any date. Identifies stars and other planets in same region. (300 and 1200 baud). **\$19.95**, with description of other programs. Overseas add \$3.00 per order for airmail.

**Eric Burgess FRAS, 14461 Frati Lane, Sebastopol, CA 95472**

-----  
**FOR SALE:** 20 each, 10-foot lengths of **40-conductor gray flat cable**. 3M#3417-3 and 3M#3418 connectors installed. **\$12** each.

**FOR SALE:** 3 each, **Track Ball assemblies** like those used in arcade games. 3" ball in 6.75"L x 6.625"W x 5"D unit. Includes encoder data sheet. You must design an interface. **\$500** each. I'll pay shipping.

To purchase, or for further information, write:

**John A. Haldeman, 1023 SW 81 St., Oklahoma City, OK 73139**

**4th TIP** - by Timothy Huang, FORTH Editor

There are thousands of reasons why I should not be writing this column. Nevertheless, here goes anyway. Hopefully, there will be more converted programmers joining the FORTH revolution through my efforts.

FORTH is a great language for serious programmers. To me it is the best, especially for the Sorcerer because of its small memory requirements. I have not found any other language that will provide the same power, speed and extensibility as does FORTH. A Pascal compiler, by comparison, can take upwards of 48K to run (and is not RomPac-able).

This column will serve as a means of exchanging ideas and working 'tools'. I don't pretend to be a seasoned FORTHer, but in the process of learning, I did pick up some useful things and developed a few myself. I'll share these discoveries in this column and hope that you will also contribute yours so that we can all benefit from each other's experiences. Hopefully, by this means, we can build up a very powerful and useful **tool kit**.

In this issue, I will direct you to some tutorial materials and classes which will help you to learn FORTH.

If you would prefer to learn FORTH in a formal academic setting, I would recommend Mr. Kim Harris' FORTH class at Arcata, CA. For further information, contact: Professor Ronald E. Zammit, Dept of Physics, Humboldt State University, Arcata, CA 95521 - (707) 826-3275

I attended Mr. Harris' class last March and found it to be an excellent course. I enjoyed it so much, that I never got to bed before 1 a.m. that whole week! If you plan to go, register early and bring your micro along with FORTH running.

If you prefer to **do-it-yourself**, then the following references should be seriously considered:

1. Fig-FORTH **Installation Manual** - \$10 (fig)
2. Fig-FORTH **8080 Assembly Language Listing** - \$10 (fig)
3. **Systems Guide to fig-FORTH** by C.H. Ting - \$20 (Mt. View Press)
4. **Using FORTH** by FORTH Inc. - \$25 (FORTH Inc.)
5. **79-Standard** - \$10 (fig)
6. August 1980 issue of **BYTE** magazine - \$ 3 (fig)
7. **Threaded Interpretive Language** by Loeliger - \$20 (fig)
8. **Forth Dimensions** (fig's newsletter) - \$12/yr. (fig)

All of the above material can be purchased from either of the following:

-the FORTH Interest Group (fig), P.O. Box 1105, San Carlos, CA 94070

-The Forth Source, Mountain View Press, P.O. Box 4656, Mountain View CA 94040 tel. (415) 962-8653

In my opinion, the articles in the August '80 issue of **BYTE**, are the best tutorial material for the very beginner.

The **Installation Manual** and the **8080 listing** are a must for those intent on getting their own fig-FORTH. For those who would like to be saved the bother of typing in this assembly listing, I'll be happy to provide it on a Micropolis Mod II diskette for \$15.

So far, the **Systems Guide** is the best description of the internals of fig-FORTH. Unfortunately, it's not a tutorial book and is printed all in upper case making it more difficult to read. Also, try not to be too critical of the author's style, his mother tongue is not English.

**Using FORTH** was published by the language inventor's company, FORTH Inc.,  
(continued on next page)

(4th TIP continued)

which publishes a version having some differences from fig-FORTH's. Hopefully, after everybody adopts the **79-standard**, this will be corrected.

The #7 item is a book for the Z-80 FORTH. Even though the author did not use the word FORTH in the title (the word **FORTH** was copyrighted by FORTH Inc.), it is every bit FORTH. If you get lost in its pages like I do, don't get discouraged. The book will become clearer with a second and third reading.

The way that works best for me is parallel reading. By making constant reference to the various publications, you'll obtain a better picture. It's said that the best way to learn a language is to use it every chance you can, every day.

Begin doing some reading. Next issue I'll cover how I got started with FORTH. Until then, may FORTH be with you!

-----  
(DAISY WHEEL GRAPHICS continued from page 81)

The necessary interface logic can be implemented using only seven 74LS-series TTL chips. My interface circuit is shown in Figure 1 (next page). This circuit generates three internal interface control signals:

1. ADDRESS SELECTED. This signal goes low whenever there is an I/O request with the printer address on lines A0-A7. When this signal is low, the tri-state buffers which transmit data to the printer are enabled.
2. STROBE ENABLE. When the printer is addressed by a Sorcerer **output** instruction, this signal gates address lines A11, A12 and A13 to the printer strobe lines (print wheel, carriage, and paper feed). The printer acts only when it receives a strobe signal. If the strobe is a printwheel strobe, for example, the printer will print the character whose ASCII code is on its data lines.
3. DATA BUS DIRECTION. This signal goes low when the printer is addressed by a Sorcerer **input** instruction. It enables the tri-state buffer which transmits printer status data back to the Sorcerer. The DATA BUS DIRECTION signal is also returned as a control signal to the Sorcerer at its DATA BUS DIRECTION pin. This is necessary!

It is also necessary to hold the Sorcerer DATA BUS ENABLE line low. In my interface, this is hard wired.

Two of the Sorcerer address lines are not used in this circuit. If the circuit was adapted to a Qume or Hytype II printer, these lines could be used for the half-step data bit and the ribbon strobe.

There are several printer control lines which I haven't shown in the diagram, SELECT READY, PRINTER, SELECT, RIBBON LIFT, and RESTORE. I've hard wired SELECT READY. However, I have chosen to make the others accessible through switches on the interface box. This has proven to be very convenient! Disabling the PRINTER SELECT line is a good way to interrupt printing (for instance, when I need to insert new paper in the middle of a listing). A **restore** button is essential if the carriage slams into its stop while the driver program is being debugged. Without a **restore** signal, the printer will do nothing unless the power is shut off and turned on again.

Two last hardware notes: First, it is a good idea for any line between the Sorcerer 50 pin connector and the printer to have some kind of a buffer between the two machines. Second, it's probably better to power the interface from the printer power supply than from the Sorcerer. With the Hytype I, this was easy since the 5 volt supply is brought out in the interface connector.

The other half of the story, the software, will follow in the next issue.

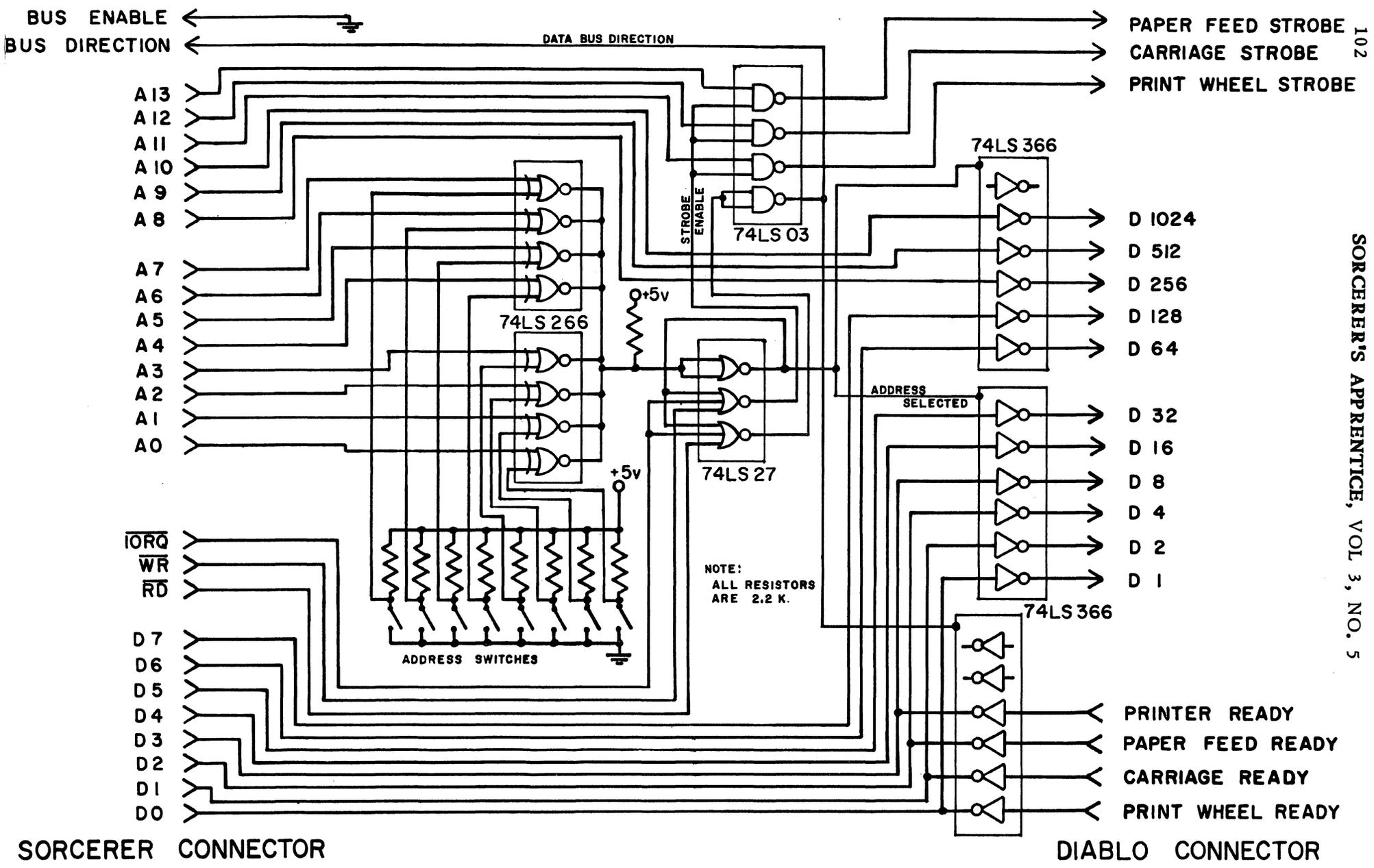


FIGURE 1. SORCERER to HYTYPE I  
HARDWARE INTERFACE

**WORD PROCESSOR TIPS - by Bradley Perkins****Limits to File Size in RAM**

If you are running the Word processor with more than 40K of Ram do not allow the size of the file in the computer to exceed 40K. Once a file gets above this size the Word Processor may bomb or it may rewrite sections of the file with random characters. There may also be a problem reading a file of this size back in. This does not mean that you cannot have a file larger than 40K, but you must read only part of it in at a time.

**Typing in the OFF END Area**

Normally when using the INDENT function, the Word Processor will not allow you to write in the area between the left screen edge and the left indented margin. Once in a while the Word Processor 'forgets' that you have indented and allows you to type in that area. If you do, there is a good chance the Word Processor will bomb. If the screen starts flickering and the Word Processor becomes unresponsive, try RESET with the ESC key down. If this fails, try a RESET and the lost file retrieval procedure explained below. **NOTE:** If the Word Processor has bombed, it is usually unwise to allow it to run rampant as text that may otherwise be saved, might be destroyed.

**Lost RAM & Cassette File Retrieval**

If you have been working on a file and hit a Word Processor bug or have accidentally hit RESET or have used a K(ill) or D(elete) command, you may be able to save some or all the file.

The Word Processor does not keep track of the size of a file. It knows where the end of the file is by placing 03 0E at the end of the text. Thus when the computer has been RESET or file has been K(illed), the file is still in RAM. When the computer has been RESET and the Word Processor is restarted or a file has been K(illed), the Word Processor places a 03 0E at address 080E, where the text begins. Thus, to retrieve a file, enter the monitor and type:

```
>EN 80E<cr>
080E: 0E 0E /<cr>
>PP<cr>
```

If you have D(eleted) some text from the middle of the file, you are probably out of luck. When the text is deleted, the Word Processor rewrites the text below over the text removed.

If you have D(eleted) text from the end, you may be able to save it. At the end of the remaining text type about 20 of any character. This will serve as a marker. Enter the monitor and start dumping the hex codes from address 080E until you spot the 20 of the same hex code. At the end of this string of same code, there should be a 03 0E. Using the ENter command, replace the 03 with a 0E.

It is possible to retrieve a Word Processor file with a CRC error or load a long file from the middle. A cassette file is made up a short header that contains the file name and blocks of 255 characters with a CRC byte. The Word Processor does not know how many blocks are in a file. It loads however many blocks you tell it to or until it finds the end of the file.

To retrieve a file with a CRC error you will need another cassette with any other Word Processor file and some method to monitor a tape while it is being loaded. First read the problem file in until you reach the tape and stop the tape immediately. Next listen to the tape to find the next area between the blocks and stop the tape. Now read in a new file header from the other tape but stop the tape right after the header and before the first block of text. Now you have 20 seconds to put the problem tape in the recorder and start it. You will loose one block of 255 characters where the error was.

Members of the **Sorcerer's Apprentice User's Group** are entitled to 8 issues of the group's newsletter, the **SORCERER'S APPRENTICE**; the services of the library; access to its on-line CP/M based Computer Bulletin Board Service; other services as they become available.

**MEMBERSHIP RATES:** USA, Canada & Mexico \$12 - bulk postage, \$20 - 1st class postage, in an envelope; all others \$18 - surface mail, \$26 - air mail.

<b>BACK ISSUES:</b>	ARESCO Source (issues 1-5)	\$ 7.50
	S.U.N. Volume I	\$10.00
	S.U.N. Volume II	\$10.00
	Sorcerer's Apprentice Vol I (1-7)	\$10.00
	Sorcerer's Apprentice Vol II (1-5)	\$10.00
	Sorcerer's Apprentice (per issue)	\$ 2.00

**Overseas orders** for back issues add \$3.00 per volume or \$1.00 per issue to cover additional postage and handling.

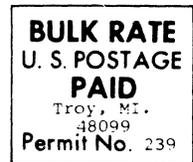
Make checks or money orders (only in US funds drawn on a US bank) payable to: **SORCERER'S APPRENTICE**.

**Commercial advertisers**, please contact Thomas E. Bassett, Advertising Manager, for advertising rates. **Non-commercial classified ads** are accepted at the rate of \$2.00 per 80 column line or part-line.

**Newsworthy items** can be submitted via the Sorcerer based CBBS (313) 535-9186, the SOURCE (TCF656), or MicroNET (70150,365), on Word Processor cassettes or CP/M Word Processor/Editor files on Micropolis Mod II diskettes (any of these preferred) or hardcopy. Magnetic media returned upon request. Hardcopy will be returned if requested and accompanied by SASE.

**SEND ALL CORRESPONDENCE TO:**

SORCERER'S APPRENTICE  
P.O. Box 1131  
Troy, Michigan 48099  
U.S.A.



**RETURN AND FORWARDING POSTAGE GUARANTEED**