

SORCERER'S APPRENTICE

PAGE 43
VOL.2
NO.5
OCT-NOV 1980

IN THIS ISSUE-

| | |
|-----------------------------------|----|
| SORCERER'S PROGRAM EXCHANGE..... | 44 |
| ANIMATED USER-DEFINED GRAPHICS... | 45 |
| MEMORY SEARCHING PROGRAM..... | 45 |
| EXATRON STRINGY FLOPPY..... | 49 |
| MAIL-MERGE MACRO..... | 50 |
| WORD PROCESSING CORNER..... | 51 |
| MACHINE CODE TUTORIAL PACKAGE... | 52 |
| USING THE DEVELOPMENT PAC..... | 53 |
| YES OR NO? BASIC ROUTINE..... | 58 |
| SCREEN BACKGROUND..... | 58 |
| MESSAGE PROGRAMMING IN ML..... | 59 |

SPELLBINDER FOR SORCERER

SPELLBINDER, a CP/M based, disk version, of the word processor "PAC" has just been released in an Exidy version, the Sorcerer's Apprentice has learned exclusively. The program has all the features of the PAC, including the touchpad editing controls.

Features included in this word processor which are not available in the PAC include all precision printer routines, title centering, dynamic line counting, global searching and printing, and full titling top or bottom, just to mention a few.

Only one disk drive is required to operate it. Disks can be switched between documents and documents can be written to more than one disk without difficulty. For those of you who write assembly language programs for CP/M, this program will set up those files for assembly.

The program will support a wide variety of print routines and print formats. All formatting is dynamic, that is, all print menu routines (abd titling routines) can be changed in text. ASCII senders are possible from text directly.

For more information on disk formats and price, write to Mentzer Electronics, 590 South Hill Blvd., Daly City, CA, 94015.

THIS IS A SPECIAL ISSUE.

READ THIS NOTICE

This issue of the Sorcerer's Apprentice is being mailed to approximately 700 Sorcerer owners. We are trying to contact all former S.U.N., ARESCO Source, Monitor subscribers; former SA readers who failed to send in their SASEs; and anyone else whose name and address we accumulated from various sources. Starting in January only paid up subscribers will receive the Sorcerers Apprentice. Don't put it off! Get your check in the mail today.

THE SORCERER'S SOFTWARE EXCHANGE-

After a period of inactivity, the exchange has received some new programs from R. Harrison, R. Leitner, and D. Bristol. Also on the list, for those of you who had trouble getting the "Mystery" program to work, is the program Ambush, S039. Use the numeric keypad direction keys, including the diagonals, to shoot down the ambushers. Send your programs and requests directly to RGR.

THE SORCERER'S SOFTWARE EXCHANGE

BY
RGR SOFTWARE COMPANY
623 MEDILL AVENUE
LANCASTER, OHIO
43130

TERMS:

\$2.00 FOR THE FIRST PROGRAM AND \$1.00 FOR EACH THEREAFTER. THERE IS NO LIMIT ON THE NUMBER OF PROGRAMS PER TAPE, ONLY TO THE LIMIT OF THE TAPE. PROGRAMS MAY BE EXCHANGED ON A ONE FOR ONE BASIS, IF THEY ARE FULLY DOCUMENTED IN REM OR PRINT STATEMENTS WITHIN THE PROGRAM.

MAKE ALL CHECKS OR M.O. PAYABLE TO RGR SOFTWARE CO.
CHECKS MAY DELAY SHIPMENT BY 3 WEEKS.

CATALOG

| NUMBER | NAME | AUTHOR | DESCRIPTION |
|--------|-----------|--------------|-------------------------|
| S001 | AMLOG | R. RUH | AMATEUR LOGGING PROGRAM |
| S002 | ZAP | R. RUH | ROBOT GAME |
| S003 | EXPENSES | R. RUH | EXPENSE RECORD |
| S004 | BILLIARDS | R. RUH | BUMPER POOL GAME |
| S005 | ALIEN | R. RUH | CAPTURE THE ALIEN GAME |
| S006 | WAMPUS | P. DOMAIN | MAZE GAME |
| S007 | HORSERACE | R. RUH | RACETRACK GAME |
| S008 | JOUST | R. RUH | MEDIEVAL JOUSTING GAME |
| S009 | LUNAR | B. BACHMAN | SPACESHIP LANDING GAME |
| S010 | GET | B. BACHMAN | USER FUNCTION DEMO |
| S011 | CRAPS | D. BRISTOR | GAMBLING GAME |
| S012 | DBASE | D. BRISTOR | DATA BASE MANAGEMENT |
| S013 | DATA | T. BASSETT | ENHANCED VERSION DBASE |
| S014 | MMIND | E. BRISTOR | TWO LEVEL GUESSING GAME |
| S015 | AMORT | T. BASSETT | LOAN AMORTIZATION |
| S016 | CHECKING | T. BASSETT | CHECKBOOK BALANCE |
| S017 | PATTERN | J. PALEVICH | PATTERN GENERATOR |
| S018 | RENUMBER | D. TRUSSELL | LINE RENUMBERING PROG. |
| S019 | MERGE | T. BASSETT | PROGRAM MERGING |
| S020 | HEXD | P. WILKINSON | HEX, OCTAL CONVERTER |
| S021 | EDIT | P. WILKINSON | TEXT EDITOR-NO INSTR. |
| S022 | GSSR | P. WILKINSON | CHILDRENS LETTER GAME |
| S023 | RGRS | P. WILKINSON | MULT. REGRESSION PROG. |
| S024 | SERCH | J. LINK | DEBUGGING PROGRAM |
| S025 | BLACKBOX | J. LINK | DEDUCTIVE REASONING |
| S026 | AUCTION | J. LINK | ART AUCTION GAME |
| S027 | CLUE | P. DOMAIN | MURDER GAME |
| S028 | LABEL | R. JOHNSON | MAILING LABEL PROGRAM |
| S029 | HORSE | P. DOMAIN | GRAPHIC RACING GAME |
| S030 | CHRIN | P. DOMAIN | CHARACTER GENERATOR |
| S031 | LUCAS | P. DOMAIN | MENTAL GUESSING GAME |
| S032 | TOUCH | P. DOMAIN | PARTY TOUCHING GAME |
| S033 | FIT | P. DOMAIN | NONLINEAR CURVE FITTER |
| S034 | SMOTH | P. DOMAIN | EXPONENTIAL ALGORITHM |
| S035 | GRAPH | R. LEITNER | PLOTS MATH FUNCTIONS |
| S036 | CORCO | P. DOMAIN | LINEAR COEFFICIENT |
| S037 | MREGR | P. DOMAIN | MULT. REGRESSION PROG. |
| S038 | SIMEQ | P. DOMAIN | SIMULTANEOUS EQUATIONS |
| S039 | AMBUSH | P. GRIMSHAW | MYSTERY PROGRAM |

ANIMATED USER-DEFINED GRAPHICS- by Ian Macmillan from the Sorcerer Computer Users of Australia newsletter

Dynamic bit control in a programmable character enables movement generated in one character cell of eight bytes to be displayed on the screen whenever that character is printed. Here is a listing that demonstrates the possibilities.

```

10 FOR X=-512 TO -504: POKE X,0: NEXT
20 FOR X=1 TO 64: PRINT "graphic shift one";: NEXT
30 REM: hit the "graphic shift one" key don't type in the words
40 FOR X=0 TO 7
50 POKE -510,2^X: POKE -511,2^(7-X)
60 NEXT
70 GOTO 40

```

MEMORY SEARCHING PROGRAM

Editor's note: This type of routine was used by the Australian User Group in Doncaster to relocate the Word Processor PAC and the Development PAC so that they could be loaded via cassette and be used at the same time the BASIC PAC was in place. Techniques to transfer files and programs between these PACs have been worked out. I have some of this documentation and I have made a copy of my relocated WP PAC. This is one of our group's ongoing projects and documentation will be available later this year.

A GENERAL MEMORY SEARCHING PROGRAM by Bryan Lewis

Here's a machine language program that will search the Sorcerer memory looking for a specific sequence of bytes. It allows wildcards, i.e.; bytes that are considered a match for anything. Let's say you are trying to understand the inner workings of the Word Processor PAC. You would like to find every time the WP PAC calls one of the Monitor routines located in the E000 page of memory. Remember that any such call will look like: CD xx E0, where the xx could be any byte. Also remember that the WP PAC is situated at C000 to DFFF. Proceed as follows:

```

GO 0
Search for what sequence of bytes? CD? E0
Start address? C000
End address? DFFF
D942: CD 1B E0
DE9D: CD 12 E0

```

The question mark is a wild card. In this instance, two matches were found: one is a call to VIDEO (E01B) and the other a call to OUTAPE (E012). The address of each call is given.

The program could be used to look for text (character strings). You just have to give it the ASCII hex codes for the string you want.

The assembly language code shows how the program works. It illustrates how to take advantage of some of the console input and output routines in the Monitor. The hex dump allows you to load the program by using the Monitor's EN command. With the program located from 0000-00FF, the GO address is 0000.

(CONTINUED ON PAGE 46)

```
*****
*                                     *
*                   SEARCH           *
*   BY      BRYAN LEWIS      9/13/80 *
*                                     *
*****
```

A general byte-searching routine for the Sorcerer. Calls Sorcerer Monitor routines to do line input and hexadecimal output.
 Written with a CP/M-based assembler, but the format is almost identical to the Development Pac's.
 Uses the byte FF hex to stand for a wild card. Thus if the user wants to search for a sequence containing the legitimate byte FF, he may get some unasked-for matches. The exact byte sequences will be reported as they're found, however, to avoid confusion.
 If the area of memory containing this program is searched, one match will always be found, with the stored key bytes.

```
WILDSYM EQU    '?'      ;USER'S INPUT SYMBOL TO DESIGNATE WILD
                          ;CARD.
WILDBYT EQU    OFFH     ;RESERVED BYTE TO BE TREATED AS A
                          ; WILD CARD, AUTOMATIC MATCH.
```

```
;;DEFINE ADDRESSES FOR THE CANNED ROUTINES IN THE MONITOR:
ADDCOL EQU    0E20FH
HEXSPC EQU    0E21CH
LINEIN EQU    0E13AH
MSGOUT EQU    0E1BAH
SCAN EQU     0E225H
CONV EQU     0E23DH
CRLF EQU     0E205H
```

```
-----
                ORG    100H
                JP     INTRO ;JUMP TO START, LEAVING STORAGE ROOM.

;
NUMBER DEFB    0          ;STORAGE FOR NUMBER OF BYTES.
KEY     DEFB    10H      ;STORAGE AREA FOR THE SPECIFIED BYTES.
CR      EQU     0DH
PROMPT  DEFB    'S','e','a','r','c','h',' ','f','o','r',' ','n',
DEFB    's','w','h','a','t',' ','s','e','q','u','e','n',
DEFB    'c','e',' ','o','f',' ','b','y','t','e','s',' ','?',
DEFB    ' ',0
STARTM  DEFB    'S','t','a','r','t',' ','a','d','d','r','e',
DEFB    's',' ','?',0
ENDM    DEFB    'E','n','d',' ','a','d','d','r','e','s','s',
DEFB    ' ','?',0

;
INTRO    LD      HL,PROMPT ;PRINT QUERY FOR KEY SEQUENCE.
        CALL   MSGOUT
        CALL   LINEIN ;GET ENTIRE LINE.
        PUSH  IY      ;PUT ADDRESS OF BUFFER IN HL.
        POP   HL
        LD   DE,KEY  ;(DE) = POINTER TO KEY SEQUENCE.
        LD   C,0     ;INITIALIZE KEY COUNT.

ENTER   CALL   SCAN  ;PARSE LINE INTO BYTES.
        JP   Z,GETADD ;GO ON WHEN NO MORE BYTES.
        LD   A,(HL) ;DID HE ENTER A
        CP   WILDSYM ; WILD CARD?
        JR   NZ,CONVERT-$ ;NO, HANDLE NORMALLY.
        INC  HL      ;YES. ADJUST SCAN POINTER.
        LD   A,WILDBYT ;CONVERT TO THE RESERVED WILD
        JR   DEPOSIT-$ ; BYTE AND GO DEPOSIT IT.

CONVERT PUSH  DE      ;CONVERT HEX INPUT TO A BYTE.
        CALL  CONV    ;CONVERT BYTE TO BINARY.
        LD   A,E      ;AND PUT INTO A.
        POP  DE      ;RESTORE POINTER.
```

```

DEPOSIT LD      (DE),A      ;STORE THE BYTE.
        INC      DE          ;GET READY FOR NEXT ONE.
        INC      C           ;COUNT.
        JR      ENTER-$     ;REPEAT.
GETADD  LD      A,C         ;STORE THE NO. OF KEY BYTES.
        LD      (NUMBER),A
        LD      HL,STARTM   ;ASK FOR START ADDRESS.
        CALL   INPUT        ;AND GET IT.
        PUSH   DE          ;SAVE IT FOR A MOMENT.
        LD      HL,ENDM     ;ASK FOR END ADDRESS.
        CALL   INPUT        ; GET IT, AND LEAVE IN DE.
        LD      BC,KEY      ;INITIALIZE KEY POINTER.
        POP    HL          ;RECOVER START ADDRESS IN HL.
        DEC    HL          ;TO GET READY.
CONT    PUSH   HL          ;SAVE MEMORY POINTER.
        XOR    A           ;CLEAR CARRY BEFORE SUBTRACTION.
        SBC   HL,DE        ;CHECK WHETHER WE'RE PAST END.
        POP    HL
        RET    Z           ;IF SO, QUIT.
        INC   HL          ;NOT FINISHED. POINT TO NEXT BYTE.
        LD    A,(BC)       ;GET NEXT KEY BYTE.
        CP    WILDBYT      ;IS THE NEXT KEY BYTE WILD?
        JR    Z,MATCH-$    ;YES, GO TO AUTOMATIC MATCH.
        CP    (HL)         ;COMPARE TO MEMORY.
        JR    Z,MATCH-$    ;IF NO MATCH. RESET KEY POINTER, GO ON.
MATCH  LD      BC,KEY
        JR    CONT-$
        INC   BC          ;INCREMENT KEY POINTER.
        PUSH  HL          ;SAVE MEMORY ADDRESS.
        PUSH  BC          ;AND KEY POINTER.
        PUSH  BC
        LD    BC,KEY
        POP   HL
        XOR   A           ;CLEAR CARRY FLAG.
        SBC  HL,BC        ;KEY POINTER - ITS ORIGIN.
        LD   A,(NUMBER)   ;COMPARE THAT TO THE NO. OF BYTES
        CP   L           ;NEEDED FOR A FULL MATCH.
        POP  BC
        POP  HL
        JR  NZ,CONT-$     ;IF NOT DONE YET, CONTINUE.
        PUSH HL          ;IT WAS A FULL MATCH. SAVE THINGS.
        LD   BC,(NUMBER)  ;BACK UP THE MEMORY POINTER TO
        LD   B,0          ;THE START OF THE FOUND SEQUENCE.
        XOR  A
        SBC HL,BC
        INC HL          ;ADJUST.
        EX  DE,HL        ;PUT INTO DE SO CAN CALL CANNED PRINT
        CALL ADDCOL      ;ROUTINE.
        EX  DE,HL
        LD  B,C          ;DISPLAY THE NUMBER OF BYTES.
        LD  A,(HL)       ;GET THE BYTE.
DISPLAY CALL HEXSPC     ;PRINT WITH SPACE.
        INC  HL          ;NEXT ONE.
        DJNZ DISPLAY-$   ;LOOP FOR NUMBER OF BYTES.
        CALL CRLF        ;NEW LINE.
        LD  BC,KEY      ;RESET POINTER FOR NEXT OCCURENCE.
        POP HL
        JP  CONT
;
;
INPUT  CALL MSGOUT      ;PRINT A PROMPTING MESSAGE.
        CALL LINEIN     ;GET A LINE.
        PUSH IY
        POP  HL
        CALL SCAN        ;PARSE.
        CALL CONV        ;CONVERT INTO BYTES.
        RET
;

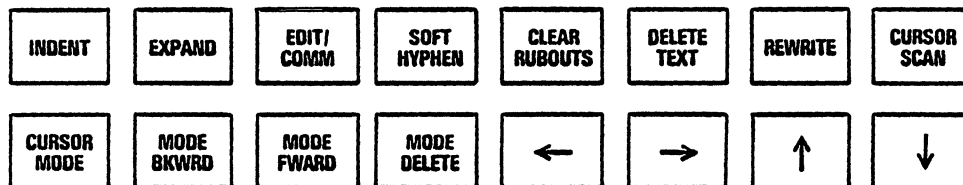
```

EXIDY SORCERER USERS

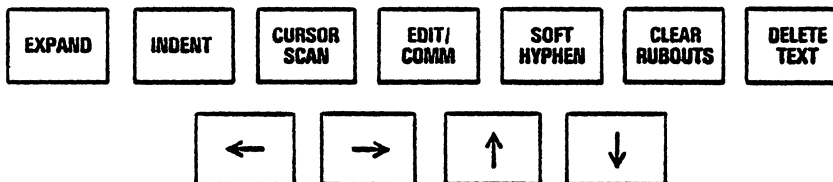
— ATTENTION —

We have custom engraved keytops for *Spellbinder* and *Exidy's Word Processing ROM Pac*.

Spellbinder set consists of the following:



Word processing ROM Pac consists of the following:



Sets are available in *Red, Blue, Black, Beige and Charcoal, with or without front engraving of symbols originally on the top of the keytop (no graphic symbols replaced).

| Pricing: | Spellbinder | ROM Pac |
|----------------------|-------------|---------|
| No front engraving | \$14.00 | \$10.00 |
| With front engraving | \$22.00 | \$15.00 |

Sets shipped UPS COD unless accompanied by check or money order.

Also available for Spellbinder are the Y and U keytops (exchange basis only), that are engraved **ENTER** and **ENHANCE** respectively. (engraving on front face)

We also stock keytops for Cherry and Keytronic Corp. keyboards and other video data terminals and computers.

ARKAY ENGRAVERS, INC.

2073 Newbridge Road • P.O. Box 916
Bellmore, New York 11710

(516) 781-9859

Dealer Inquiries Invited

* We recommend RED for maximum visibility.

AN INVITATION FOR ARTICLES

We've received requests from subscribers on several different topics not yet covered in the Newsletter. We invite those with experience in the following areas to submit articles:

- Interfacing light pen, joy sticks or bar code reader
- Sound/Music generation
- Implementing a disk system
- Home control
- Interfacing to instruments

AN INVITATION TO VENDORS

We invite vendors of Sorcerer hardware/software to make submissions for review. Contact us regarding our 'review' policies.

CONTACT- Ralph LaFlamme, Secretary, Sorcerer's Apprentice
P.O. Box 1131
Troy, MI 48099

or via the SOURCE TCF656 or MicroNET 70150,365

THE EXATRON STRINGY FLOPPY - by Ralph LaFlamme

We've had many requests for information on implementing the Exatron Stringy Floppy on the Sorcerer but haven't received any information from anyone who has successfully done so. I tried to get an S-100 version running on my Sorcerer II several months ago but was not successful. If anyone has had better luck please let us know.

This mass storage system has many advantages: 1) high transfer rate (7,200 baud single density and 14,400 baud double density); 2) high reliability; 3) single density version has loading compatibility with versions made for other computers (e.g. TRS-80); 4) operating system similar in function to a CP/M allowing an exchange of programs between machines (e.g. TRS-80 to Sorcerer); 5) an extensive library of programs; 6) an extended basic; 7) an extensive user group; 8) courteous and helpful vendor.

Since my system has 48K of RAM and I wanted to use my Word Processing Pac, I had no free areas of memory for the Stringy Floppy controller card. The controller comes addressed to B000 (the top 4K of RAM). With Exatron's technical assistance, Bob Hageman (our librarian) and I worked for several weeks trying to use available 'phantom' signals to make this area of RAM 'disappear' to the monitor on cold boot. Time limitations kept us from exploring other possibilities. Exatron then took the unit back without question.

One of our club members is now working with Exatron in developing a Stringy Floppy dedicated to the Sorcerer. If anyone else is interested in this project and has the necessary hardware and software background please write us describing your qualifications and experience and how you would implement it. Please do NOT contact Exatron. They have asked us to pre-screen people. We will keep you advised of developments.

MAIL-MERGE MACRO FOR THE WORD PROCESSOR PAC-

I have received several letters from users who could not get the mail-merge macro to work (page 48 of the WP PAC manual). Hunter Marion, 278 Saxer Ave., Springfield, PA 19064 used the routine successfully and writes the following:

"Until recently I was unable to get the mail-merge macro to work. It turned out the examples in the manual were misleading. Mike Wiesenberg of Exidy furnished new copies of the examples which were easier to understand and with his help I made a mail-merge that included 120 members of my high school class and it worked like a charm.

It turned out my main problem was spacing in the macro. In the example on page 50 there should be no space after the Graphic 6 characters. That accounted for most of my problem. The other correction was an error in the reformatting statement at the beginning of the letter on page 49: the statement should begin with a Graphic 5 not a Graphic 3." (Word Processor Graphic Characters are shown in fig. 5, page 44 of the manual.)

Steve Guralnick, who has used the routine, adds another warning: the delimiter (Graphic 8) at the end of the letter is essential!; otherwise the macro will run right through the letter and shut down.

Hunter Marion's copies of the examples in his manual match mine. My manual is the second edition. Later editions may have the corrected examples. The new example of the mail-merge sample letter starts like this:

```

Ø/// Ø / 1f ---- As submitted by EXIDY and works perfectly
Ø55 // 1Ø / 1f -- This one works best for my typeface.
Ø1f
Ø2   Ø3f
Ø4, Ø5 Ø6f
Ø7f           f = Carriage return as appears on screen
Ø           Ø = Graphic 6 mark
Ø
Dear Ø8:f
Ø

```

We at Exidy would like to thank you for your interest in our Sorcerer Computer. We feel that our computer is a serious tool

The end of the letter looked like this:

```

Let me extend our thanks again, Ø8, and also my envy for
living in Ø4.f
Ø
Ø

```

Thank you,f

```

Ø
Ø
Ø

```

X.E. DeInguet
Product Manager,f
EXIDY Inc.f

```

Ø
Øf
-f

```


THE WORD PROCESSING CORNER

9

Steven Guralnick
15 Southgate Ave., Suite 246
Daly City, CA 94015

It is indeed a pleasure to be able to write today about SPELL BINDER and to share this program with all of you.

It is hard to know where to start with this incredibly powerful software. For those of you who have the pak, I will start out by saying that you will have no difficulty in running SPELL BINDER. The basic editing commands will seem instantly familiar. It's what follows after that that makes SPELL BINDER so super.

For one thing, you get the printer routines that so many of you have called or written about. Text centering is provided and it's easy to work with. Titling is much more versatile than before. You can specify a top title or a bottom title, with page number or words, or both. You can even specify where the page number goes and you can control that automatically, depending on odd or even page numbers!

The line counting is complete in this version. If you mix single and double spacing in the text, SPELL BINDER counts all the spaces between the lines in determining the length of a page.

The disk routines are simple and, best of all, you can run on one drive. All that is necessary to switch disks is to switch the disk and query its directory with a simple command. Then, text can be written to the new disk. Backups are easy and any file written to disk with the same filename as a previous write automatically gives a ".BAK" extender to the earlier file.

SPELL BINDER supports all the precision routines of our Diablo, including shadow printing, bold facing, proportional right edge justification, proportional printing, etc.

I am debating writing a regular column on this program and a lot will depend on how many of you will be buying it. If you are interested, let me know.

See you soon!

HOW'S THE NEWSLETTER DOING? by Larry Kobylarz, Editor

We have many interesting projects and articles lined up for future issues. Many contributors from the S.U.N., Source, and Monitor newsletters have joined the Apprentice team and are reporting about their experiences and projects. Bryan Lewis, Terry Calvert, Roger Harrison, Leslie Zatz, John Hill and Hunter Marion have all been in touch recently. Howard Arrington, Steve Guralnick, and Tom Bassett keep a steady flow of newsworthy material coming in. Larry Stempnik and Ralph LaFlamme are my "local" correspondents.

Jack MacGrath sent in a tape with the Sargon Chess program adapted for the Sorcerer. He also sent in a copy of the new Exidy BASIC manual which replaces "A Short Tour of BASIC". It is no longer a short tour as it covers all of the commands in more detail. A complete review will be included in a future issue.

Andre Babin and Heinz Spiess have sent in a list of their projects, which include adapting A.Crowe's Assembler; adapting the BASEX compiler; development of a text editor, disassembler, music editor, plotting program, joystick interface, light pen interface, and an interrupt driven real time clock. An article about their disassembler will be in the next issue.

It has taken a couple of months to make or renew these contacts but now the letters seem to be flowing in and I'm looking forward to putting this material together in Volume 3 of the Sorcerer's Apprentice.

MACHINE CODE TUTORIAL PACKAGE- from System Software, 1 Kent St., Bicton, 6157, Australia. Price \$24.95.

This package contains eight programs each of which is about 8k long. It is designed to show the BASIC programmer what to do after typing BYE. Each lesson will occupy the user for 1 to 2 hours reading the text, entering ML routines interactively, examining memory and running the examples. I found these lessons educational, easy to follow and entertaining. Once you load it into the machine you don't want to quit.

Some of the subjects covered include: Monitor Commands, Z-80 Instructions, RAM and ROM, Source and Object Codes, Video and Keyboard Routines, Input and Output Vectors, Memory Maps, Video and Screen RAM, Graphics and DMA, Cursor Control, Monitor and BASIC Work Areas, Monitor Routines, Storage and Linking of BASIC Programs, Parallel and Serial Interfacing, Cassette Routines, Sound Generation, No-Stop Keyboard Input, Disabling CTRL C and ESC and more.



MENTZER
ELECTRONICS

590 SOUTH HILL BLVD.
DALY CITY, CA. 94044
(415) 584-3402

NEW! SPELLBINDER Disk Word Processor for the Sorcerer, uses single function keys like the pac. Many new commands. **INTRODUCTORY SPECIAL \$350.00**
Regular price \$495.00

- Competitive prices offered on all EXIDY hardware as well as MICROPOLIS Disk Drives, GODBOUT S-100 Cards/Systems, MORROW Disk systems, QUME Daisy Wheel Printers, and Centronics Printers.
- 3M DISKS 5 1/4" and 8" single sided, single density \$33.00 box of 10. Call for other types.
- All hardware is burned in and tested before shipping.
- Contact us for a quote on your hardware needs.
- California residents add 6% Sales Tax. UPS Shipping Charges added to all orders.

THE DEVELOPMENT PAC

(#2)

LARRY STEMPNIK
433 MINNESOTA
TROY, MI 48084

Since the first column I have been trying to decide what this column should cover. It will not be a general purpose discussion on Z80 Assembly language programming. There are many books available which cover the subject far better than I could. More on this later.

What it should cover are aids in using the Development Pac, errors in the Pac, and using it to write better programs. It could also include more examples on getting started with the Pac. Any comments on topics to include or exclude from you readers would be appreciated.

REFERENCE INFORMATION

Since the last column I have discovered a number of helpful articles and books. The July,1980 issue of the Sorcerer Computer Users of Australia has a long excellent article on how to get started using the Development Pac. If you still have not been able to use your Pac after reading the first column, this article might help you.

The best book I have seen for learning assembly language programming is "HOW TO PROGRAM THE Z80 " by Rodney Zaks. Most Radio Shack stores carry it or you can order it from SYBEX, 2344 Sixth St., Berkeley, CA 94710. 80-US magazine has articles every issue about Z80 programming. The October,1980 issue of SEMCO DATA BUS had a good article by Dave Welsh on how to avoid creating problems for yourself when writing Z80 programs. It's worth reading.

FUTURE PROJECTS

Some ideas to be covered in future issues of this column are techniques to smoothly transfer files from the Word Processor to the Development Pac. Source files should be created with the Word Processor because of the editing flexibility and then transferred to the Development Pac. Also, BASIC disassemblers should be able to load the disassembled code back into the Development Pac or Word Processor file to create a new assembly with modifications.

PROBLEMS

Does anyone have a program for using one bit of the parallel port for serial input and output? I would like to operate at 110 and 500 baud without hardware modifications. Any baud rate other than 300 or 1200 using the serial port involves changing either the UART or the UART clock frequency. A software solution would be preferred over a hardware solution.

Any comments,suggestions,or examples related to the Development Pac would be appreciated. Just sent to the above address to the attention of Larry Stempnik.

ARRINGTON SOFTWARE SERVICE

9522 LINSTOCK, BOISE, IDAHO 83704
 BY: BILLY TAN AND BOB STAFFORD

SUPER-X EDITOR - SUPER-X is designed as a screen editor and source file utility for use with the Sorcerer's Basic Rom Pac. It supports single stroke commands and direct-mode execution. A flashing inverse cursor signifies that SUPER-X is ready and active.

To edit a line, first list it on the screen and then move the cursor to the location to be modified. Now you can overtype, insert, delete, or truncate. After editing, hit RETURN and the edited line will be updated in memory.

Just look at all of the powerful features that are packed into SUPER-X which make your Sorcerer easier to use.

| | |
|---------------|--|
| CTRL D | Delete character under cursor. |
| CTRL ^ | Insert subsequent displayable characters. |
| CTRL T | Truncate end of line from cursor position. |
| CTRL RUB | Erase line from screen. |
| TAB | Tab cursor multiple of 8 columns. |
| SHIFT TAB | Return cursor to column 1. |
| TAB column1 | Skip over line number. |
| CTRL R | Disable SUPER-X, return to Basic. |
| CTRL B | Disable SUPER-X, execute Basic statement in cursor line. |
| CTRL TAB | Enable SUPER-X after CTRL R or CTRL B. |
| CTRL X | Exit from SUPER-X altogether. |
| #L n1 n2 | Selectively list lines n1 through n2 inclusive. |
| #L n1 | List line n1 only. |
| #S n1 "..." | Search for token string from line n1 to end. |
| #S "..." | Search for token string from start of program. |
| #R | Recover lost program from NEW, CLOAD and RESET!!! |
| @ n1 n2 | Number screen from n1, increment n2. Auto line numbering. |
| @ n1 | Number screen from n1, increment 10. |
| #U | Request following utilities. |
| R n1 n2 n3 n4 | Renumber: new number, increment, block start and end. All default combinations. R n1 n2 does entire program |
| F n1 | Find address in memory of line n1. |
| D n1 n2 | Delete lines n1 through n2 inclusive. |
| M n1 | Merge program from tape unit n1. |
| C | Compact program by removing spaces. |
| CR | Compact program by removing spaces and REM statements. |
| CTRL M | Execute Monitor command from Basic direct-mode!!! |
| RETURN | Update line containing cursor. |

Arrington Software Service is honored to have been selected to market this exceptional utility package. This is one of the most desirable and useful programs ever to be offered. Many of you have asked for the above features. HERE IT IS!

G R A P H I C S P A C K A G E

COPYRIGHT (C) 1980 BY HOWARD ARRINGTON

TO LOAD -- Type **BYE** (Exit to Monitor)
 >LO (Load graphics routines called VECT7 or VECT3)
 >PP (Exit back to Basic)
 CLOAD (Load demonstration program called DEMO1 and RUN it)

TO USE -- The first variables in your Basic program must be X, Y, and PG in that order. Arguments are passed to the machine language routine through these variables by their position in the symbol table, not by their name. Calls to the graphic routines are through the USR() function. Establish the call address with POKE 260,0:POKE 261,S/256 where S=13056 for 16K version, and S=29440 for 32K version. Features are selected by the first letter in the USR() argument.

USR(G) -- Use this at the beginning of your program to initialize the screen variables and define the graphic characters. Initialization defines the bottom left corner dot a coordinate (0,0). The screen is organized as 128 columns by 90 rows with each dot referenced by a coordinate (column#,row#). The column numbers must be between 0 and 127 inclusive and is passed in variable X. The row number must be between 0 and 89 inclusive and is passed in variable Y. The routine truncates fractional parts of X and Y and also performs a modulo 127 on them.

USR(P) -- Access the plot routine in the mode selected by the value of P. When P=3 a straight line is drawn from the last reference point to the point now referenced by the X and Y coordinates. P=1 is the opposite of P=3 in that a straight line is erased. P=0 moves the last referenced point marker to the new (X,Y) point, which is analogous to lifting your pencil and moving it to a new place on your paper. After each USR(P) call the new (X,Y) point becomes the last-referenced point for use as an end point of the next line drawn. The routine does not change X, Y or P values.

USR(D) -- Shift the screen down one character cell, ie. three dot rows. Top row filled with graphic spaces.
USR(U) -- Shift the screen up one character cell, ie. three dot rows. Bottom row filled with graphic spaces.
USR(L) -- Shift the screen left one character cell, ie. two dot columns. Rightmost column filled with graphic spaces.
USR(R) -- Shift the screen right one character cell, ie. two dot columns. Leftmost column filled with graphic spaces.
USR(M) -- Move the screen to memory starting at the address in variable PG. Example: PG=S-2000).
USR(S) -- Move memory to the screen starting at the address in variable PG. Opposite of USR(M).
USR(I) -- Inverse video the graphic characters above ASCII 192 inclusive on the page referenced by the address in PG.

USR(F) -- Fill memory or the screen with the number of characters specified by variable X, starting at the address in PG, with the ASCII character stored in variable F. Example -- filling the screen with graphic spaces: X=1920, PG=61568-65536:F=192:Z=USR(F). Think of both the screen and a memory page as a 1920 character block.

USR(O) -- Logically 'OR' the graphic characters on the page referenced by PG with the graphic characters on the screen. If either the page cell or the screen cell is not a graphic character, then the screen remains unchanged. The logical 'OR' means that a dot appearing on the page or on the screen will make a dot in that position on the screen.

VARIABLE PG -- PG stores the starting page address used by USR(P), USR(M), USR(S), USR(F), USR(I) and USR(O). You may have as many 1920 character pages in memory as you have space for. PG=61568-65536 references the top of the screen.

GRAPHICS SET -- The graphics character set is all combinations of 6 dots in a cell. The characters occupy ASCII 192, which is the graphic space, through ASCII 255, which is the cell with all six dots showing.

SINGLE DOTS -- Plot single dots by making the end points of a line the same point. Example: P=0:Z=USR(P):P=3:Z=USR(P)

COMMON ERRORS -- Calling the USR() function with an invalid first letter generates a ?FC ERROR message. The arguments may be more than one letter such as Z=USR(LEFT). Be sure to have X, Y, and PG as the first three variables used in your program. Be sure to have PG contain the screen address for plotting on the screen. Funny plots will occur if X and Y coordinate values exceed their ranges. Use the graphic space of 192 and not the regular space of ASCII 32 for USR(I) and USR(O) applications. Arguments P and F need values. Best of all, look to the demonstration program for examples.

DEMONSTRATION -- The Basic demonstration program has 4 minutes of superb figures, plots, bar charts, circles, stars, X-Y-Z function plots, and screen motion. This software really shows off the Sorcerer's graphic capabilities!

MUSIC SYSTEM

COPYRIGHT (C) 1980 BY HOWARD ARRINGTON

HARDWARE -- Assembled A/D converter connects to parallel port. Simply connect an amplifier and speaker such as your stereo to listen to music. A \$10 amplifier from Radio Shack works, but low notes saturate the 2 inch speaker.

MUSIC GENERATION -- Machine language routine generates four part harmony of true frequencies over a 4 and 1/2 octave range. The routine easily interfaces with your Basic programs to add sound effects and music. Well documented.

EDITOR -- System includes a screen editor that displays the two musical staves upon which musical notes are placed. Quickly create song data files using the following features. Data file creation is done graphically on the screen.

SAMPLE SONG -- A peppy song is included to demonstrate the fanastic possibilities. Just load the programs and listen.

| | | | | |
|---------------|----|------|-----------------------|---|
| TO USE | -- | Type | BYE | (Exit to Monitor) |
| | | | >LO MUSIC | (Load music generation routine) |
| | | | >LO FANF | (Load song data file) |
| | | | >PP | (Exit back to Basic) |
| | | | CLOAD MUSIK | (Load the MUSIC SYSTEM'S screen editor) |
| | | | RUN | (Execute editor, screen displays staves and keypad definitions) |
| | | | PRESS 'PLAY SONG' KEY | (Press the 0 key on the 10-keypad and the demonstration song will play) |

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 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.

CASSETTE -- Save and load song data files on cassette with the Monitor's >SA and >LO commands. The data file starts at 1700 hex and its length is dependent on song length.

EASY TO USE -- Customers have often written saying they are interested in having music on their Sorcerer, but are hesitant to buy music software because they 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 the demonstration and PIANO PLAYER. Creating song data files is done graphically. Just copy sheet music onto the screen.

PIANO PLAYER

This program is compatible with the above music system and adds delightful graphical animation of a high resolution piano player playing your songs. 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 a medley of favorite Christmas songs to brighten the coming season. Being both graphical and musical, this program is a crowd pleaser twice over. (Note: MUSIC SYSTEM necessary.) This program will make a super Christmas gift.

| | | | | |
|----------------|-----------|---------------------------------|-------------|-----------------------------|
| TO USE: | >LO PIANO | (Load in piano player) | >LO MUSIC | (Routine from MUSIC SYSTEM) |
| | >LO XMAS | (Load in this or any song file) | CLOAD MUSIK | (Editor from MUSIC SYSTEM) |

MUSIK MODIFICATIONS: 130 CLEAR 500 1490 POKE 260,0:POKE 261,L-2:X=USR(0):GOTO 470 (New entry address)

ARRINGTON SOFTWARE SERVICE

9522 LINSTOCK, BOISE, IDAHO 83704

BACH MUSIC FILE - Exceptional rendition of Bach's "Jesu, Joy Of Man's Desiring" for MUSIC SYSTEM.
 OTHELLO - Challenging algorithm and special input method make this game fun to play against computer or friend.
 DISASSEMBLER - Disassembles machine language code into standard Zilog mnemonics.
 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.
 CROSS REFERENCE - Complete listing of variables and the line numbers where used. Also references all GOTO, GOSUB, etc.
 ARTILLERY - Requires GRAPHICS PACKAGE. Opponents fire cannon balls at each others castles. Screen resolution 128x90.
 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.
 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.
 1 VOICE MUSIC - Includes file editor for music entry and 6 songs of exceptional quality and effort. I'm impressed.
 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 - General: Add, Create, Delete, Edit, List, Load, Merge, Printer, Query, Report, Sort, Tabs, Write.
 MAZE RACE - Race against the clock to find your way through a random maze. Various levels of difficulty.
 BASEBALL - Very nice screen display with graphical characters batting and running bases. Select what is pitched, etc.
 SCREEN GENI - Utility to improve upon Basic's PRINT statement. Gives effect of print at, and print using control.

My apology for such brief descriptions of excellent software. Write for further details and watch for user reviews. 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.

| | | | | | | |
|--|---------|------|--|---------|-----|-----|
| <input type="checkbox"/> MUSIC SYSTEM | \$40.00 | BMUS | SORCERER SIZE: | 16K | 32K | 48K |
| <input type="checkbox"/> PIANO PLAYER | \$15.00 | MS | RETURN ADDRESS: | | | |
| <input type="checkbox"/> BACH MUSIC FILE | \$ 5.00 | S | | | | |
| <input type="checkbox"/> OTHELLO | \$10.00 | BG | | | | |
| <input type="checkbox"/> DISASSEMBLER | \$10.00 | BU | | | | |
| <input type="checkbox"/> QUBIC | \$10.00 | BG | | | | |
| <input type="checkbox"/> QUICK EDIT | \$10.00 | MU | <input type="checkbox"/> 1 VOICE MUSIC | \$15.00 | BMS | |
| <input type="checkbox"/> EDITOR FOR BASIC | \$15.00 | MU | <input type="checkbox"/> SUPER-X EDITOR | \$30.00 | MU | |
| <input type="checkbox"/> CROSS REFERENCE | \$10.00 | MU | <input type="checkbox"/> CASSETTE FILES | \$15.00 | MU | |
| <input type="checkbox"/> GRAPHICS PACKAGE | \$25.00 | BMU | <input type="checkbox"/> BLACKJACK | \$10.00 | BG | |
| <input type="checkbox"/> ARTILLERY | \$10.00 | BMG | <input type="checkbox"/> SPACETREK (32K) | \$15.00 | BGS | |
| <input type="checkbox"/> SPACE INVADERS | \$20.00 | BMGJ | <input type="checkbox"/> DATABASE SYSTEM | \$25.00 | MU | |
| <input type="checkbox"/> MUSICAL HORSERACE | \$10.00 | BMGS | <input type="checkbox"/> MAZE RACE | \$10.00 | BG | |
| <input type="checkbox"/> JAIL BREAKOUT | \$10.00 | MGS | <input type="checkbox"/> BASEBALL | \$15.00 | BG | |
| <input type="checkbox"/> CHESS 'BRUCE' | \$20.00 | MG | <input type="checkbox"/> SCREEN GENIE | \$15.00 | MU | |

KEY: B-Basic M-Machine language U-Utility G-Game S-Sound J-Joystick or keyboard

Please don't let this advertisement go to waste. Share it with your Sorcerer friends, and use this order form to place your order today!!!! This advertisement was printed on my Hewlett-Packard 2631A serial printer, using the Exidy Word Processor Pac and my VISTA 5 1/4 inch double density disk drive. The software offered is excellent -- if you have have hesitations write for further information.

YES OR NO?- by Tom Bassett

```

100 REM:-----
110 :
120 REM:      HERE IS A SHORT ROUTINE THAT IS USEFUL
130 REM:      IN THOSE PROGRAMS THAT CONTAIN A NUMBER
140 REM:      OF PROMPTS THAT MUST BE ANSWERED 'YES OR NO'.
150 REM:      IT WILL SAVE MEMORY IF THERE ARE MORE THAN TWO
160 REM:      OR THREE OF THEM.
170 :
180 REM:      LINES 300 AND 310 ARE FOR DEMONSTRATION.  IN A
190 REM:      REAL PROGRAM THEY WOULD ACCOMPLISH SOMETHING.
200 :
210 REM:      THOMAS E. BASSETT
220 REM:      FRANKENMUTH, MI 48734
230 :
240 REM:-----
250 :
260 PRINT CHR$(12)
270 Y$="": PRINT: INPUT "Answer Yes or No"; Y$
280 GOSUB 340
290 ON OK GOTO 300, 310, 270
300 PRINT: PRINT "Your answer was Yes": PRINT: GOTO 270
310 PRINT: PRINT "Your answer was No": PRINT: GOTO 270
320 END
330 :
340 IF Y$="" THEN 370
350 OK=ASC(Y$): IF OK=89 OR OK=121 THEN OK=1: RETURN
360 IF OK=78 OR OK=110 THEN OK=2: RETURN
370 OK=3: PRINT "Invalid response": RETURN

```

SCREEN BACKGROUND- by Larry Stempnik

A background grid on the CRT screen can be useful to avoid typing lines longer than 63 characters or to check formatted DATA or PRINT lines. To use this routine RUN 60000 whenever you need to refresh the background. By using double spaced printing (POKE 322,0) listings will have a background grid.

```

60000 S$=" 5 0 5 0"
60001 REM:      S$ IS MADE UP OF 4 SPACES, A 5, 4 SPACES, A ZERO
60002 REM:      THEN THE SEQUENCE IS REPEATED.
60010 FOR X=1 TO 32: PRINT S$;S$;S$;"1234": NEXT X
60011 REM:      ADD A SPACE AFTER 1234 IN "1234 " TO SKIP
60012 REM:      EVERY OTHER GRID LINE
60020 PRINT CHR$(17): REM: SEND CURSOR HOME

```

NOTE: If you want a different background try a period or graphic one in place of the 5's and zeroes. If you don't need the column markers but want an end of line indicator replace the 5's and zeroes with spaces and change the "1234" to " I" (space, space, space, I).

MESSAGE DISPLAY PROGRAMMING ROUTINE FOR ASSEMBLY LANGUAGE PROGRAMS-

The following routines by Bob Stafford were included in an article by him in the Sorcerer Computer Users of Australia newsletter.

My greatest gripe with the Development PAC is the lack of TEXT or DEFM pseudo-op to allow entry of ASCII strings for messages. If you want to include the messages in the assembly language program, you must use either the DEFB or DEFW to enter the strings one or two characters at a time which is both time consuming, space consuming, and frustrating.

I have written a small machine language program which will create a message file in a format that allows easy use of the monitor SENDLINE routine for up to 255 separate messages in a program. In this context, a message could be a single character or up to screen full of information. This has the advantage of making the message display programming easy and also keeping the message file separate from the program, thus saving space in the source buffer.

| ADDR | OBJECT | SOURCE CODE | COMMENTS |
|------|--------------|--------------|--------------------------------|
| 0000 | 21 00 03 | LD HL,0300H | ;Load starting address |
| 0003 | 3E 0C CLR | LD A,12 | ;Load A with CHR\$(12) |
| 0005 | 36 00 | LD (HL),0 | ;Start message with zero |
| 0007 | 18 19 | JR OPC-\$ | ;Clear screen |
| 0009 | CD 18 E0 MS1 | CALL OE018H | ;Input from kbd |
| 000C | 28 FB | JR Z,OUT-\$ | ;Wait for character |
| 000E | 23 | INC HL | ;Increase data pointer |
| 000F | FE 1B | CP 27 | ;Char= ESC? |
| 0011 | 28 1A | JR Z,OUT-\$ | ;Go to OUT if it is |
| 0013 | FE 0B | CP 11 | ;Char=TAB? |
| 0015 | 28 EC | JR Z,CLR-\$ | ;Start new message if it is |
| 0017 | FE 7F | CP 127 | ;Char=RUB? |
| 0019 | 20 06 | JR NZ,MS2-\$ | ;Continue if not |
| 001B | 2B | DEC HL | ;Move data pointer to |
| 001C | 2B | DEC HL | ;character before last |
| 001D | 3E 08 | LD A,8 | ;Load A with screen erase |
| 001F | 18 01 | JR OPC-\$ | ;Rub-out if same line |
| 0021 | 77 MS2 | LD (HL),A | ;Store character input |
| 0022 | CD 1B E0 OPC | CALL OE01BH | ;Display character on screen |
| 0025 | FE 0D | CP 13 | ;Char= CR? |
| 0027 | 20 E0 | JR NZ,MS1-\$ | ;Continue if not |
| 0029 | D6 03 | SUB 3 | ;Set A= LF |
| 002B | 18 F5 | JR OPC-\$ | ;Do LF |
| 002D | 2B OUT | DEC HL | ;Set data pointer to last zero |
| 002E | EB | EX DE,HL | ;Put last address in DE |
| 002F | CD E8 E1 | CALL OE1E8H | ;Print end address |
| 0032 | C9 | RET | ;Return to monitor |

NOTES ON USING THIS ROUTINE

1. Machine code is relocatable.
2. It stores the message file starting at 0300H. To change this, alter the values in the second and third bytes.

(CONTINUED ON PAGE 60)

3. Most messages will begin with at least one CR. There is no need to include a LF after a CR since the monitor routine SENDLINE automatically does this. The routine sends a LF to the video monitor so that the display will look like the message to be output.

4. At the end of each message, press the TAB key. This will both insert a zero at the end of the message and CLEAR the screen for the next message.

5. At the end of the file, after the last TAB, press the ESC and the end address of the message file will be displayed at the top of the screen as a four digit hex number.

6. To save the file on tape use SA MESS 300 ADDR, where ADDR is the address displayed.

7. To erase the previous character entered, use SHIFT-RUB. This will erase the character in the file and if it is on the current line of the screen it will also be erased there. Be careful with this as it is possible to erase back before the start of your file.

TO USE THIS FILE WITH YOUR ASSEMBLY LANGUAGE PROGRAM

Include in your program the following routine which will find the message you specify and print it on the screen:

| | | | |
|----------|------|---------------|--------------------------------|
| 21 yy xx | OPMS | LD HL,MSAG | ;xxyy=start of message file |
| AF | | XOR A | ;Clear A |
| BE | OPM1 | CP (HL) | ;Find a zero |
| 23 | | INC HL | ;Next character |
| 20 FC | | JR NZ,OPM1-\$ | ;Find end of message |
| 10 FA | | DJNZ OPM1 | ;Find specified message |
| CD BA E1 | | CALL OE1BAH | ;Output message using SENDLINE |
| C9 | | RET | ;Return |

Also include as the last statement of the Source Code:

```
MSAG EQU $
```

and load the message file immediately after the loaded program. Then anywhere in your program that you want to output a message, load it's number (1-255) into B and call OPMS; e.g.: To output first message-

```
LD B,1
CALL OPMS
```

The registers affected are the A, B, and HL; with the HL pointing to zero at the end of the message output on RET. Note that if you load B with a number greater than the number of messages in the file, you will get rubbish until a zero is found.