

MICROPOLIS USERS GROUP

MUG Newsletter # 9 - April 1981

\*\*\*\*\*

ANOTHER NEW FORMAT

MORE DATA, LESS SPACE

Yes, here we go again. The third format change of the year. For those of us who want to make sure you're getting your money's worth, the new 6-page format allows room for 6% more characters than the old 10-page format. The old page size for text was 7x10 on an 8 1/2 x 11 page. Therefore, there were (7") X (12 char/in.) X (60 lines/pg) X (9.3 pages), or 46,872 characters. The new format, before reduction, is (5.1") X (10 char/in.) X (92 lines) X (2 columns) X (5.3 pages), or 49,735 characters.

Additional increases in actual text space also result from having the blank lines which separate paragraphs now use 50% of the space they use to. Short code lines also use less space.

The purpose of all this is to reduce the number of pages reproduced, and the weight. Five pages in an envelope, going airmail overseas, costs \$1.20. The newsletter can now be expanded by 33% and still go for \$.80. I also think the layout is more professional. Comments are welcomed.

PRODUCTION PROCESS ALSO CHANGED

Another change, not as obvious to the reader, is the method of production of the newsletter. I bought a new program called TEXTWRITER from Organic Software. In combination with Systemation's EDIT/S, it makes a pretty fair wordprocessing system, and is being used to produce the newsletter. One can use LINEEDIT, or even BASIC, if EDIT/S isn't available. Best of all, I can work with either a high-speed or a letter-quality printer.

Up to this point, I've retyped all incoming correspondence that gets printed in the newsletter. Although I still welcome hand written letters, I'd now prefer that correspondence meant for publication come on disk in LINEEDIT format. Line length isn't important. TEXTWRITER takes care of re-sizing lines.

To make it worth your while, I will consider the submission of an article an acceptable trade for a copy of a library volume. An article doesn't have to be totally ready for publication. I'll edit it if you wish - maybe even if you don't wish. But to qualify for the trade, I do mean an article similar to those in this month's newsletter, and not a letter of comment or questions.

.....

SYSTEMATION'S DISCOUNTS EXTENDED

Systemation's Bob Zale tells me that they will extend their discount program an additional two months. So, through the end of May, you can still get 10 to 15% off Systemation products. See newsletter #7, pages 8 & 9, for details. Take your discount from the prices listed there, which are retail. Bob isn't raving about the response so far. I wonder why people aren't taking advantage of this offer? Systemation puts out some great utilities. In all honesty, I couldn't function without them.

.....

THE MUG LIBRARY

The library is officially in business. We have one full MOD II disk. As stated before, if you send me a disk containing a new program for the library, and \$3 (\$5 overseas), I'll send you back your disk with a current library volume. If you want a copy

without sending a program, send \$15 (\$17 overseas) to cover the costs of the disk, postage, packaging, and my time.

All dollars must be in U.S. funds. To ease the problem for those of you who don't have access to US dollars, I'll accept disks at a barter rate of \$5 each. Please, no stamps. I can't use them. That is, say you're from Canada and want one library volume without sending a program. Send four disks. I'll keep three and send you one back. Include enough disks to meet or exceed the library cost. I'll credit you with the excess.

If you want multiple disks, send multiple programs. Only one free diskcopy per submitted program. I'm sorry to say that this applies to MOD I, even though there's less than half the data on each disk. Actually, it's much more difficult for me to make MOD I copies because DISKCOPY doesn't, at the moment, do MOD I copies on my MOD II system.

I have enough material to fill a second MOD II disk, so there will be a new listing next month. My criteria for selection of this set was based on variety. The remaining programs (and some new ones) are just as good, but I only have so much time for documenting what I'm doing. To tell you the truth, it's a real chore keeping everything straight.

By the way, don't get upset by seeing Tom Hogan's Basically Speaking Software copyrights when you list his programs. Tom is now editor of InfoWorld and has gone out of the software business. He graciously gave the MUG permission to include these programs in our library.

Listed below are the contents of Revision 00 of the MOD II Library Disk 01. MOD I Disk 01-A contains MONEYPONE through VISLETTER, MOD I 01-B contains LOADGO through TRANS, and MOD I 01-C (not full) contains the remainder of the MOD II 01 programs.

.....

MUG MOD II Library Disk 01, Revision 00

Name	Type	Rev	Author/Description
====	====	===	=====
MONEYPONE	BAS	00	Hogan, T. Load and run for control of the following 4 programs for home money management.
LOANS	BAS	00	
NETWORTH	BAS	00	
CHECKBOOK	BAS	00	
BUDGET	BAS	00	
GRADEBOOK	BAS	00	Hogan, T. Run in conjunction with the following program.
REPORTCARD	BAS	00	
AMORT	BAS	00	Smith, B. Amortization, with its description in the following file.
AMORT-TEXT	SRC	00	
POSTMAN2.0	BAS	00	Hogan, T. Mail-list routine.
DEPREC	BAS	00	Shapiro, J. Depreciation.
AMORT.S	BAS	00	Shapiro, J. Amortization.
INVOICE	BAS	00	Shapiro, J. Invoice generation.
FORMULA1	BAS	00	Shapiro, J. This and the following program are variable rate mortgage calculators.
FORMULA2	BAS	00	
STATISTICS	BAS	00	Hogan, T. Control program for the following 5 statistical programs.
MMS-MEAN	BAS	00	
MMS-LINCOR	BAS	00	
MMS-T-TEST	BAS	00	
MMS-ANOVA	BAS	00	
MMS-COVAR	BAS	00	
VISLETTER	BAS	00	Anders, M. A nice example of the rudiments of a BASIC word-processor.
LOADGO	SRC	00	Micropolis. Allows automatic execution of BASIC application program from initial boot.
POKE.VIDEO	BAS	00	Maschino, G. Some elements of

			graphics manipulation for a memory mapped video in this and the next program.
POKE.NRS	BAS	00	
DATES	BAS	00	O'Brien, D. Converts a numeric date to day of week.
RLTIME.DOC	SRC	00	Rowland, H. Real-time clock routines description, with assembly language source in next file.
RLTIME.SRC	SRC	00	
START	BAS	00	Anders, M. Load and run for control of the following 9 game routines.
DATA	DAT	00	
KEY	BAS	00	
CRAZYTALK	BAS	00	
BATTLESHIP	BAS	00	
LUNARLAND	BAS	00	
NERVES	BAS	00	
SHUTTLE	BAS	00	
IGUESS	BAS	00	
YOUGUESS	BAS	00	
SPACETAXI	BAS	00	
SPACCAPTUR	BAS	00	Maschino, G. Game - Capture the Klingon.
BIGNUM	BAS	00	Rudow, B. Game which converts numeric input to text.
TICTACTOE	BAS	00	Smith, B. Game - with the following file being description.
TICTACTEXT	SRC	00	
ROMAN	BAS	00	Mac Kenzie, M. Converts arabic to roman numerals.
ONT-PAY	BAS	00	Mac Kenzie, M. A Canadian payroll program.
GENSORT	BAS	00	Burkhardt, E. Merges a set of unsorted records into a sorted file.
TRANS	BAS	00	Rudow, B. A BASIC filecopy utility.
BARCOM	SRC	00	Barnum, B. Assembly language Communications routine.
CALENDAR	BAS	00	Mac Kenzie, M. Generates a calendar.
NAMESORT-1	BAS	00	Mac Kenzie, M. BASIC bubble sort of names via DATA statements.
NAMESORT-2	BAS	00	Mac Kenzie, M. Same as above, except names via Keyboard input.
POKE.1	BAS	00	Maschino, G. Combination of sorts and graphics for memory mapped video in this and the following two programs.
POKE.2	BAS	00	
POKE.3	BAS	00	

.....

SYSTEM PROGRAMMING TIPS

by the Micropolis S/W Engineering Group

In response to requests for internal (and external) information on various parts of the PDS system, we, the Software Engineering Group at Micropolis, would like to make a deal with you.

Micropolis' corporate policy is such that the sharing of some of the information you would like is forbidden due to the possibility of increased support costs. In addition, the available resource in the engineering group is very low at this time (and always.)

However, we would like to help our users in problems with the PDS system.

So here's the deal. Questions to the Micropolis Engineering Users group should be addressed to the Micropolis Users Group. We will answer any questions we can through the Users Group. Any information that we supply beyond clarification of the manual must stay unsupported. What that means is that under no circumstances should support problems with that information be directed to Micropolis directly. If our management sees effort being expended to support information that we supply to

the Users Group, they will not allow this exchange to continue.

So we will try to answer questions that appear in the Users Group newsletter, if they are addressed to us, with a general reply each month. If we miss a month or decline to answer a question, give us a break, we still love you!

Except for special circumstances we will only provide information pertaining to the 4.0 and 4.1 releases of the PDS system or the 1.0 release of the OSM system.

**QUESTION:** Is there some mod to LINEEDIT that I could make to have it generate 250 character records?

**ANSWER:** No, there is no possible mod to LINEEDIT as LINEEDIT uses the @SAVEDATA and @LOADDATA routines in MDOS. However, MDOS could be patched such that the sequential write logic would only generate 250 character records. In MDOS 4.0 the address of some words which control the length of records written sequentially is 1982 and 198A. In addition, at addresses 1A0A and 1A0D the low and high order bytes of the desired length are stored. We haven't actually tried this info out. Tell us if it doesn't work.

**QUESTION:** Is there a way to read assembly language (or BASIC) files when using an assembly language routine called by a BASIC program?

**ANSWER:** Yes there is a way, however it is complicated. BASIC completely overlays the MDOS section of PDS. This section contains all of the @ level routines for doing file management. Inside the RES section is a much more primitive file management modular which is used by both MDOS and BASIC. We can not (right now) divulge the interface to this "kernel". Maybe a little later. This is touchy information.

By the way, in OSM all file management capabilities are available to all programs in either assembly or BASIC including assembly language functions called by BASIC. In addition, the calling conventions of the file manager have been completely changed (and improved). Unfortunately this means that existing assembly language code for file management must be rewritten. That's progress! We don't mean to be flip. A lot of soul searching and hair tearing went in to the change.

**QUESTION:** How about the entry points and parameter passing requirements for the arithmetic routines in BASIC?

**ANSWER:** Even if we gave them to you, you would not be able to use them very well. Not the least of the problems is that any errors that occurred in the processing would cause the program to crash. The mathpak is kind of built into the program parser in a messy way. Run time errors all bottom out instead of cleanly returning error codes like MDOS routines.

**QUESTION:** I know that CIN (etc.) is the logical routine, CDIN is the physical routine, but what is the purpose of @CDIN?

**ANSWER:** The I/O sub-system of PDS is split into several levels. The @CDIN level provides the facilities that handle the ASSIGNment directives (i.e. printer echoing.) The original intent of the table driven structures (@CIOTABLE and @LIOTABLE) was to allow overlaying of I/O drivers. The extra layer of logical vs physical I/O routines (CIN and CDIN) was added later to simplify the configuration process.

**QUESTION:** Is there any assembly language listing for RES, MDOS, BASIC or any of the rest of the programs in the PDS group, available? Even bits and pieces.

**ANSWER:** NO. Not even bits and pieces. SORRY!

**QUESTION:** Do you have a routine for converting one

type file to another?

ANSWER: We aren't quite sure what this question means: 1) We do not have a routine. 2) Conversion of a BASIC program file to a LINEEDIT file could be programmed in assembly language. The main problems would be: the difference in line number range, the expansion of the keyword tokens to ASCII text and the reformatting of the result text line to the LINEEDIT format.

The LINEEDIT format is explained in section 4.4.26 of the PDS manual.

The BASIC format is as follows: Each line begins with up to 5 bytes that hold the line number in ASCII. The remainder of the line contains the line's text followed by a hex FD. The line's text can contain ASCII characters mixed with keyword tokens. Keyword tokens are 8th bit high bytes (except for FD and FE). You can discover the mapping by writing a record to a data file with a one character string containing an 8th bit high character and then reading it back into a string. BASIC will detokenize the character. After the terminating FD byte of the last line (if any) the program is terminated with an FE.

#### IN CONCLUSION

Just to justify our existence, here is a freebee. You may have noticed that there is no good way to set the load address in a record through MDOS. It must be frustrating to you since MDOS and the ASSMbler obviously do exactly that. The secret is the existence of two undocumented routines in MDOS: @RFILESECTOR and @WFILESECTOR. The interface to these routines is as follows:

@RFILESECTOR on Entry: -file number in B register  
-record number in DE registers

on Return: -Record address in HL registers  
-record is read into the filebuffer

@WFILESECTOR on Entry: -file number in B register  
-record number in DE registers  
-record address in HL registers  
-record already buffered in the filebuffer

on Return: -error code in A and carry flag

```
@RFILESECTOR EQU 1A8E
@WFILESECTOR EQU 1AD3
```

Once again let us restate our position. Do not call or write Micropolis for support on any of the above information. If you have problems with it, write the Users group and we will try to reply.

.....

#### BASIC PROGRAMMING TIPS

by Don O'Brien  
13085 Sky Park Drive, Omaha NE 68137

#### THE FMT STATEMENT

I have tried some things on my own and found that they work. One is the fact that you may use other characters in a FMT statement, such as a '/' or a '-'. With this concept, one may express a date which has been input as a real number D (010381) by using:

```
PRINT FMT(D,"Z9-99-99")
or
PRINT FMT(D,"Z9/99/99")
```

This yields "1-03-81" and "1/03/81", respectively. The manual seems to cover this, but not too clear-

ly. This has been very useful for me. I have seen some routines which took several lines of code to accomplish the same results.

Another feature which I have found helpful is the fact that a FMT statement can be assigned to a string variable.

```
D$=FMT(D,"99-99-99")
```

This allows one to use the RIGHT\$, MID\$ and LEFT\$ statements to separate the string into separate pieces.

#### UNUSED MEMORY

I have also found that there is an area of memory which is not used by MDOS or by BASIC. This is 00H to, but not including, 006AH. I use this area to store the date, but it could be used for anything you wanted to store from program to program. Even re-booting will not destroy the data in this section. I can change the data any time I like.

#### APPLICATIONS

In application, consider the following code:

```
020 IF CHAR$(PEEK(3))<>"/" OR CHAR$(PEEK(6))<>"/" G
OSUB 300
.....(Go on with normal program).....
290 ! CLEAR SCREEN AND CURSER POSITIONING FOR INTER
TUBE CRT
300 PRINT CHAR$(12);CHAR$(27);CHAR$(89);CHAR$(44);C
HAR$(52);
310 PRINT "***** GOOD MORNING!! *****"
320 PRINT
330 INPUT "PLEASE ENTER TODAY'S DATE (MMDDYY) FOR U
SE IN TODAY'S PROGRAMS";X
340 PRINT
350 PRINT TAB(20);"THANK YOU! HAVE A NICE DAY!"
360 FOR I=1 TO 500
370 NEXT I
380 D$=FMT(X,"99/99/99")
390 FOR I=1 TO 8
400 POKE(0+I)=ASC(MID$(D$,I,1))
410 NEXT I
420 RETURN
```

Include these lines in your highest level program, that is, the one that is always run first and which probably contains a menu which selects operating options. The first time line 20 is executed, the TRUE condition will cause the call of subroutine 300, thereby setting the date, which will remain across all program loads.

If you want the capability to change dates, include the following lines and an option to call them.

```
600 FOR I=1 TO 8
610 D$=D$+CHAR$(PEEK(0+I))
620 NEXT I
630 PRINT CHAR$(12);CHAR$(27);CHAR$(89);CHAR$(44);C
HAR$(32);"TODAY'S DATE IS ";D$
650 INPUT "DO YOU WANT TO CHANGE DATES";Y$
660 IF LEFT$(Y$,1)="Y" GOSUB 900
670 PRINT CHAR$(12)
680 RETURN
900 INPUT "ENTER THE NEW DATE (MMDDYY)";X
910 D$=FMT(X,"99/99/99")
920 FOR I=1 TO 8
930 POKE(0+I)=ASC(MID$(D$,I,1))
940 NEXT I
950 RETURN
```

All application programs should include lines 600-620 to read the date out of low memory. You, of course, then PRINT D\$ whenever you need a date.

.....

MORE BASIC PROGRAMMING TIPS

by Burks Smith, of DATASMITH  
P.O. Box 8036, Shawnee Mission KS 66208

YES/NO DETERMINATION

One of the more common routines used in applications programs asks the operator a question and expects to get a "YES" or "NO" answer. A typical service routine for this type of question must check whether "YES" or "NO" was entered and execute an appropriate part of the program in response. Additionally, the service routine must be able to detect an answer which is neither "YES" nor "NO" and prompt the operator to re-enter.

Because this is such a common routine, it is normally encountered in many places in a program and it is advantageous to keep it as short as possible while still performing all the required tasks. A routine that doesn't adequately check for a "garbage" operator response may behave unpredictably and result in a program bug.

The following routine is extremely compact and has all the desirable features:

```
100 A$=" ":INPUT "ENTER YOUR ANSWER";A$
110 ON INDEX("YN",LEFT$(A$,1))+1 GOTO
    120,200,300
120 PRINT "INVALID RESPONSE, PLEASE RE-
    ENTER":GOTO 100
```

---  
200 ! Line 200 handles the "Y" response.

---  
300 ! Line 300 handles the "N" response.

In all, the routine takes just three lines: one to ask the question, one to evaluate the response, and one for an error message. The response evaluation line is the one that does the work and is made up of three components. (1) An ON...GOTO statement; (2) An INDEX function, and (3) A LEFT\$ function. This is how it works:

The user's answer is assigned to the variable A\$. This is a string variable, so anything from the keyboard is valid. Line 110 is handled by BASIC in the following sequence:

(1) The LEFT\$(A\$,1) function isolates the first character of the user's response. Only this single character is evaluated so any string beginning with "Y" is considered to be a "YES" response and any string beginning with "N" is considered to be a "NO" response. Thus, "Y", "YES", "YEAH", "YUP" or "YUGOSLAVIA" are all considered to be "YES" responses.

(2) The INDEX function returns the position of the first character of A\$ in the string "YN". If A\$ begins with "Y", the INDEX function returns 1; if A\$ begins with "N" the INDEX function returns 2. If A\$ begins with anything but "Y" or "N" the INDEX function will return a zero, because it won't be in the string "YN". After the INDEX function does its work, we will have either 0, 1, or 2 to work with.

(3) Next, 1 is added to the number we have obtained above to make the number 1, 2, or 3. This is used by the ON...GOTO statement to determine which of three line numbers the program will GOTO, depending on what A\$ was. The first line number points to the error message, the second line number points to a routine to be executed if the response was "Y" and the third line number points to a routine to be executed if the response was "N".

USEFUL BUGS?

There is one possible pitfall in using this routine, due to a bug (?) in the Micropolis INDEX function. It seems that the "index" of a zero length string in any other string is always 1. Therefore, if A\$ is an empty string, the routine will act as if it were a "Y". It is for this reason that A\$ is set to a single space before the

input statement. If the operator simply hits RETURN in this case, the routine will treat it as an invalid response.

One man's bug is another man's feature, so the INDEX "bug" could be used to advantage if desired. For instance, an empty string response could be taken as a "default" answer, in this case "YES". If the order of the key string were changed to "NY", then "NO" would be the default for an empty string.

.....

A FILE-OPENING SUBROUTINE

The code listed below is a subroutine used for the repetitive job of opening a data file on drive 1. Instead of retyping the code each time I write a program, I merge this routine into my code and call it when needed. Joel Shapiro has spoken to me about his theory that programming is a lot easier if one has a subroutine library. While Joel will hopefully expound on the subject in future issues, the concept seemed reasonable enough for me to implement some library routines. This one is used on both the MUG membership and software vendors disks, neither of which have been seen by anyone as yet. But that's another story.

The relatively high line numbers (30000) are used to keep the routine on top of the "work" code. As stated in previous newsletters, when BASIC does a GOTO or GOSUB, it starts at the lowest line number and searches through the program until it finds the proper line. Therefore, routines which are executed only once should be numbered above the "work" code so that time is not wasted by searching through them.

The "O\$(4)" in line 30015 clears the screen. O\$(4) is initialized by calling Dave Land's configuration routine. See page 2 of newsletter 8.

This routine tries to keep you from aborting a program in mid-run. Line 30018 checks to see that the drive is on-line. Line 30048 allows you to change disks if the current one doesn't have the file you want. Line 30054 allows you to re-enter an incorrectly spelled file name. The variable ERR\$, line 30066, will contain the proper error message for the current failure.

```
30000 !
30003 ! Subroutine for displaying contents of disk
30006 !           & selecting file for operation
30009 !
30015 PRINT O$(4)
30018 OPEN 8 "1:DIR" ERROR 30063
30021 CLOSE 8
30024 PRINT TAB(10);"THE FOLLOWING FILES ARE AVAILA
    BLE:"
30027 PRINT
30030 DISPLAY "1:DIR"
30033 PRINT
30036 PRINT "If desired file is not listed, insert"
30039 PRINT "another disk, type 'X', press RETURN."
30042 PRINT
30045 INPUT "Enter Name of File Desired:";F$
30048 IF F$="X" OR F$="x" GOTO 30015
30051 F$="1:"+F$
30054 OPEN 1 F$ ERROR 30063
30057 RETURN
30060 !
30063 PRINT
30066 PRINT "*****";ERR$;*****
30069 PRINT "Correct Problem, Press RETURN to Conti
    nue."
30072 PRINT
30075 INPUT R$
30078 GOTO 30015
```

.....

AN 'INKEY' ROUTINE

Several people have asked how to implement the INKEY function contained in some other BASICs. Micropolis BASIC does not support a function which

gets a single keyboard character response without pressing RETURN. Since it is a nice feature, I have implemented the following assembly language routine as a substitute.

```

0000 05F0 0000 * INKEY 03/21/81
0000 0010 LINK 'SYSQ1'
0000 0020 ORG 04EH
004E 3E 03 MVI A,3
0050 32 A0 01 STA @INBUFF
0053 3E 01 MVI A,1
0055 32 A1 01 STA @INBUFF+1
0058 32 A2 01 STA @INBUFF+2
005B CD 7B 07 CALL @CIN
005E 78 MOV A,B
005F 32 A3 01 STA @INBUFF+3
0062 CD 7F 07 0110 CALL @COUT
0065 CD 0B 00 0120 CALL @CCRLF
0068 C9 RET

```

An example of implementation is the following BASIC program.

```

10 DEF FAA=16R45 5F0
20 PRINT "INPUT A SINGLE DIGIT NUMBER? ";
30 A$=FAA(1)
40 IF A$<"0" OR A$>"9" PRINT "INCORRECT RESPONSE!":
   GOTO 20
50 PRINT A$
60 END

```

The subroutine prints the response at line 110 and the program prints it at line 50. The redundancy was used to show the alternative. Remove either line when you use it.

If you are looking for a YES/NO response, line 40 of the program might read:

```
IF (A$<>"Y" AND A$<>"N") PRINT .....
```

You will have to load the object file before running the BASIC program. Borrowing Don O'Brien's idea, I've put it in low memory, so it will stay there as long as your computer stays on. If you don't want to assemble it, use the ENTR 004E command of MDOS, followed by the object code listed above. See page 4-4 of the Micropolis manual for use of ENTR, and newsletter #5 for a discussion of construction of an object file. If you do assemble it as, for instance, INKEY.L, I see at least three ways to get it in the system. You could LOAD "INKEY.L" after you get BASIC up. You could insert the line

```
05 LOAD "INKEY.L"
```

in my BASIC program. Or you could insert the line

```
415 LOAD "INKEY.L"
```

in Don's program. Using the latter method would cause the file to load only if the date had not been previously initialized.

The relatively unique feature of this concept is that neither Don's or my program has any dependance on what computer configuration you have. Input/Output ports, memory size, type of machine monitor, memory mapped or terminal video - nothing makes any difference. These are truly universal applications which make use of the common MDOS structure and interfaces which we all have.

....

#### SOFTWARE FOR SALE

##### SYSTEMATION'S UNPROTECT

Systemation has released UNPROTECT, their second CP/M utility. UNPROTECT restores the original source code for any program saved in the protected format of Microsoft BASIC-80. Available on MOD I or MOD II, and IBM compatible 8" disk formats, for \$70 (\$63 to MUG members). CP/M 2.0 or later is required. Contact Systemation, inc., P.O. Box 75, Richton Park IL 60471 (312) 481-2420.

.....

#### SYNTAX'S TAX PROGRAMS

Syntax Corporation's SHORTAX program (mentioned in newsletter #6) is now priced at \$500 rather than \$250. This program is designed for the professional financial and tax advisor rather than for the general public and they have spent a lot of additional money on developing quality documentation.

However, Syntax will provide a free copy of TAXMATCH and FINCAL to MUG readers who purchase a copy of SHORTAX. SHORTAX is a fully documented program with an easy to read user manual. The TAXMATCH and FINCAL programs are functional but are not yet documented for commercial purposes. When they are suitably documented, the price will have to go up. The SHORTAX manual is available for \$15.00.

The G/Ledger program is way behind schedule and Syntax isn't ready to deliver any copies at this time, unless the buyer wants to take it on an as-is basis for a discounted price. (Negotiable)

By the way, the SHORTAX program is available from Dave Land at the Computer Center, and Syntax would like to send a manual to any other MDOS system dealers for their consideration. Contact Vern Jacobs at Syntax Corporation, 4500 W. 72nd Terrace, Prairie Village KS 66208 (913) 362-9667.

.....

#### LETTERS

##### BASIC & GRAPHICS

Buzz,  
I just received my first copies of the newsletters, and am eagerly looking forward to the next MUG's issue.

Just thought I would let you know my inner feelings about owning a Vector Graphic System B. I have had the unit since last June, using the Peachtree Software with some modifications, running under CP/M. The hardware has been trouble free, with only minor software difficulties. All in all I am very satisfied, though I wish the system would operate a little faster.

My problem has been understanding many of the formatting commands of Micropolis BASIC. The manual that came with the System B does not show examples. Most books written about BASIC seem to be for the TRS-80 or Apple or Pet, but none are written for Micropolis BASIC. I called Micropolis and told them of my problem as a new owner and beginner programmer and they told me I should buy their manual, which I did. I received the same manual I have from Vector.

I would like to try my hand at some simple graphics. There is no mention of graphics in the manual. Can anyone help me? Please advise me what I might purchase to better understand Micropolis BASIC.

Richard M. Herz  
74-76 Wythe Ave., Brooklyn NY 11211

.....

Richard,  
I try to print explanations of some BASIC directives each month. Obviously, this is a random approach and isn't particularly useful to the new programmer. You are not alone in your request for instruction.

There is no book on Micropolis BASIC. Books that teach variations of Microsoft BASIC (such as the TRS-80) should be useful, as Microsoft and Micropolis BASICs are quite similar.

Anyway, I'll give formal instruction a try, starting next month. Readers who have suggestions on how to present material on BASIC and Graphics

PAGE 6

should call or write me and give me their thoughts. Time is always a problem, too. If there are any of you willing to write a monthly column on these subjects, on assembly language, RES, MDOS, and BASIC physical structures, CP/M, or any other subject - please contact me.

While I'm on the subject, I want to thank - very, very much - the people who have been contributing letters and articles. Please continue to do so. I'd like those of you who haven't sent anything yet to give some thought to putting down on paper your discoveries, accomplishments, and problems. Remember, this is a newsletter, not a slick magazine. Information interchange is the name of the game.

.....

CLASSIFIED

FOR SALE: Two Micropolis 1033-2 dual disks, without controllers; \$600 each or make offer. Ben Evers (714) 565-4704 (days), 469-3629 (nights), or write P.O. Box 307, Spring Valley CA 92077.

WANTED: Source of double-sided 16-sector 5 1/4 inch disks. Source for protective mailing jacket for 5 1/4 inch disks. Buzz Rudow

.....

NEXT MONTH

Status of Systemation's Compiler  
Contents of Library Disk 2  
A look at CP/M  
More MDOS BASIC & Assembly Language routines

.....  
04/01/81

FIRST CLASS MAIL

FIRST CLASS MAIL

Subscription (August through July) rates:  
U.S., Canada, Mexico; \$12/year; Other, \$25/year  
Mid-year subscribers receive current year's back issues.  
Published monthly by the MUG

MICROPOLIS USERS GROUP  
BUZZ RUDOW, EDITOR  
604 SPRINGWOOD CIRCLE  
HUNTSVILLE AL 35803  
(205) 883-2621

FIRST CLASS MAIL