

**GET YOUR FORTH NEWSLETTERS NOW! (Editorial)**

It's renewal time for the MMSFORTH Newsletter! This is Newsletter #6, the final bimonthly issue of your first \$10.00 subscription. A subscription to Volume 2 - five more issues to complete 1981 - is available to our licensed users now. Send us a \$10.00 check, along with your name, address, phone and MMSFORTH Serial Number. (No charge cards for just \$10.00, please, but you may combine it on a larger order.) MMS aims to have information worth that amount within EACH issue. And I think we will!

The Post Office has raised some of its rates, and so has MMS. Our minimum shipping/handling charge is now \$2.00, and our overseas mail rate is back up to 20%. We still refund overpayments, so add some cushion when you are not sure.

The Forth Interest Group has asked us to tell you that its newsletter has gone up in price: \$10.00 for Volume 1 or Volume 2 of 4th Dimensions, \$12.00 for the present year (\$13.00, \$13.00, & \$24.00 for overseas orders). Did you listen to us and get FIG's Volume 1 when it was only \$5.00?

-- A. Richard Miller, Editor 4th Class

**NEW AT MMS:****A BOOK FOR DESIGNING FORTH PROGRAMS:**

Are you learning Forth, but still unsure how to design your programs to take best advantage of its new capabilities? MMS is pleased to offer "Program Design and Construction", by David A. Higgins. This useful book does not discuss Forth, but does teach an easy technique for the design of structured and modular programs. We find it to be an excellent introduction to the special kind of programming best realized in the MMSFORTH environment. In lieu of a specific programming language, the book itself uses a simple and powerful program structuring notation called Warnier-Orr diagrams. A final section is written with examples in BASIC, but by then these will be easily translated into MMSFORTH by the reader. Far easier, in fact, than in the less natural BASIC!

"Program Design and Construction" is available now at MMS, for \$8.95 plus our minimum \$2.00 shipping/handling charge. Massachusetts customers please add the necessary 5% State tax.

**PERIPHERAL TALK****MMSFORTH PATCH FOR EXATRON STRINGY FLOPPY:**

On a custom basis, we've been running MMSFORTH on the tiny Exatron Stringy Floppy wafers for a year or so. This small and quick-loading peripheral is popular with computer hobbyists who are not yet ready to invest in disk drives, and it also is an excellent professional choice for certain applications. Now our Canadian MMSFORTH User Group organizer, Kalman Fejes, can provide you with his own \$14.95 ESF patch (on a wafer) to merge your MMSFORTH System Cassette aboard. We haven't yet seen his routine, but we know the MMSFORTH/ESF combination can be a fascinating one. Contact Kal to order or for further information.

**SETTING FORTH (for beginners and others)****DATAHANDLER CUSTOM MODIFICATIONS: OVERVIEW**

THE DATAHANDLER, our database management system in MMSFORTH, is an excellent general purpose system. With a few modifications for any given task, it can become a dream come true!

MMS is very busy customizing THE DATAHANDLER for a wide variety of clients. So far we have produced dedicated versions to do specific tasks involving professional mailing labels, inventory, payroll, order entry and sales analysis, repair logging, and more. All out-perform their BASIC counterparts (often spectacularly so!) and were delivered in less time with less expense. In one of its more versatile roles, MMS has even modified THE DATAHANDLER to replace a fleet of IBM keypunch machines with one-disk Model I's for preprocessed high-volume data entry into a minicomputer system!

Other programmers as well as MMS now provide custom installation and modification of THE DATAHANDLER. Typical prices are \$500 including all software and consulting time for a simple system, \$1,000 for a moderately complex project. At \$140, MMSFORTH and THE DATAHANDLER probably offer 90% of the final job in completed and modular form. That offers a good profit for a

good Forth programmer, and could offer a full- or part-time business for you. For the client it's a good price, early delivery, and a lot of user satisfaction as well.

Because so many of you have expressed interest in this as a personal or commercial activity, MMS will explain modification procedures and key internal words for THE DATAHANDLER in this and upcoming issues.

**DATAHANDLER CUSTOM MODIFICATIONS: TECHNIQUES**

This is the article many of you have been waiting for, a first trip through the MMSFORTH source code of THE DATAHANDLER to teach it a few new tricks - and to see how MMS goes about doing such an operation.

**THE PROJECT AND THE TOOLS:**

We will be using MMSFORTH V1.9, its on-board full-screen editor, and THE DATAHANDLER V1.1. We will add the MERGE routine published in MMSFORTH Newsletter #1, and the MAKE-LAST routine from NL#5. In addition, we will write a new Forth routine to create custom headings on our report tables. It's a good opportunity to add and debug any MMSFORTH and DATAHANDLER fixes we've mentioned in these Newsletters, too!

This will be a complicated operation. But if you have mastered our prior beginner projects, you should be able to keep up with us.

**THE GAME PLAN:**

Because THE DATAHANDLER is an efficient in-memory system, it allocates to the use of its maximum-sized files just about any RAM that isn't already used by Forth and itself. This DATAHANDLER File Buffer is allocated in 1K (1,024-byte) chunks of RAM, so if we are lucky we may find additional room for our mods within the existing "last" chunk of the present program. If not, we must either trim some from the existing DATAHANDLER commands, or limit the file buffer size in order to gain RAM space for our added programming. During development, we will limit file size even further in order to run the Full-screen Editor, the .S (nondestructive stack print-out) routine, etc. After programming, testing and debugging, we will remove these extra routines and again make the file buffer size as large as possible. That's the plan, now we need only fill in the details!

**DISK ORGANIZATION FOR THE DATAHANDLER:**

THE DATAHANDLER source diskette we supply consists of Forth source blocks defining additional routines for various functions, then a files directory block and finally a series of blocks dedicated to the files themselves; three sample files are delivered on the diskette.

Layout of the delivered diskette is quite simple: Blocks 1 and 2 are delivered with temporary routines for adding your MMSFORTH system to this disk or a backup thereof. Your MMSFORTH system then will occupy Blocks 0-9. Blocks 10 and 11 are a disk directory like the one on your MMSFORTH system diskette, but for THE DATAHANDLER. Blocks 12 through 15 bring some necessary MMSFORTH optional routines (strings, double-precision, etc.) over for this application. THE DATAHANDLER itself begins on Block 16 and ends on Block 47. Actually, its normal mode ends on Block 45; the next two blocks define DIR-INIT, which is a special routine for creating custom DATAHANDLER Files Directory Blocks. Block 48 is a standard Files Directory Block. From there on up we have provided three sample files and then a bunch of empty, formatted blocks, on up through Block 86, the final block on a standard 35-track diskette.

(See Memory Map and Diskette Maps on Pages 2 and 3.)

Don't just take our word, see for yourself. Boot up your MMSFORTH system, then swap in your DATAHANDLER disk, enter O EDS and keep hitting the + key to scan through and look. While in the full-screen editor, be sure to avoid other actions which could re-write the block! Better yet, keep a write-protect tab in place during this first look.

If you don't have a merged DATAHANDLER/MMSFORTH source diskette like the one described above, merge one now according to your DATAHANDLER instructions. Then make a BACKUP on which we will take the rest of this trip.

**LOAN THE DATAHANDLER AN EDITOR:**

It is possible to do all the editing for our new routines before loading and running THE DATAHANDLER program. But our debugging will be greatly facilitated if THE DATAHANDLER is up and





MMSFORTH MODIFICATIONS

**MAKE-LAST FIX:**

Our thanks to Nick Reinhardt of Lexington, Mass. for reporting an error-producing typo in our last issue. In the first line of the MAKE-LAST definition, a "@" character was accidentally deleted. Fetch it back, so the line reads:  
 : MAKE-LAST ONE IF IRECORD DUP RECORD-# @ 1- IRECORD DUP @

**MERGE MYSTERY:**

Our MERGE command for THE DATAHANDLER (Vol.1:2 of this Newsletter) does fine in actual situations, but we have detected a strange effect when demonstrating it by "doubling" an existing file; i.e., MERGEing it with itself. If the original file was small enough the doubled file has twice its number of records, as expected. However, SAVEing and then MERGEing the file a second time only produces three times the original number of records, instead of the expected four! The apparent problem only occurs when MERGEing the same data twice from fewer than three file blocks, so that the newly-saved file doesn't get re-read into the two block buffers from diskette (because THE DATAHANDLER thinks it's already in the block buffers).

If this paradox disturbs you, just modify the MERGE routine by inserting the words FLUSH ERASE-CORE on Line 10 between MERGE and CLS.

**NEW JKL BLOCK:**

The TRS-80 Model III is incompatible with the way the present MMSFORTH JKL routine, on Block 34, uses PAD. Also, MMS has been exercising the many options of the Epson MX-80, an excellent and inexpensive new dot-matrix printer which we are using and selling. It has a mode in which it accepts TRS-80 graphics characters, as does the Okidata Microline-80. We prefer its normal mode which has many other options, but in this mode each TRS-80 graphics code must be increased by 32 decimal in order to print the proper character.

Our replacement Block 34 solves both problems, elegantly!:

```
BLOCK : 34
0 ( JKL-TO-PRINTER ROUTINE )    DECIMAL
1
2 : ALFA OVER + SWAP ( STARTING ADDRESS # OF BYTES ALFA
3 DO I C@ DUP 127 > OVER 192 < AND ( TRS-80 GRAPHICS CODE?
4 IF 32 + ( <-- CHANGE THIS LINE: "IF 32 +" FOR EPSON
5 ( PRINTER, "IF" FOR OKIDATA, ELSE "IF DROP 46".
6 ELSE DUP 32 < OVER 191 > OR
7 IF DROP 46 THEN ( IF UNPRINTABLE, SUBSTITUTE A PERIOD.
8 THEN ECHO
9 LOOP ;
10
11 : JKL PRINT 15360 16 0
12 DO DUP 64 -TRAILING ALFA CR 64 +
13 LOOP DROP CR CR CR CR CR CR CRT %CONT ;
14
15 ' JKL INTRP !
```

**STRETCH YOUR STRINGS WITH IN\$ AND \$COMPARE:**

Earlier versions of IN\$ may be identified as a single line of Forth source code which appears on Block 30 Line 1 of the MMSFORTH System, and on Block 13 Line 4 of THE DATAHANDLER. These versions are not suitable for strings longer than 64 bytes. Substitute this two-line version of IN\$ to handle up to 255 bytes at a bite,

instead:

```
: IN$ " ? " PAD 255 EXPECT 1 PAD TOKEN DROP HERE C@ HERE +
PAD HERE C@ + HERE C@ 1+ -MOVE PAD ;
```

To find room in Block 30, you can Delete Block 29 Line 1 and then move \$XCHG to Block 29 Line 15. (To be neat, relabel the 0-lines accordingly.) We made room in Block 13 by moving the \$.R definition up alongside \$.L - dirty, perhaps, but effective.

\$COMPARE, defined on the same blocks as IN\$ in both MMSFORTH and THE DATAHANDLER, also will need modification to compare strings longer than 128 bytes. Find the five-line definition beginning CODE \$COMPARE, and change its second and third lines to read as follows:

```
C INR B INR BEGIN HL INX DE INX C DCR =0 IF B DCR 0 HL LXI
BC POP =0 IF PSH THEN HL DCX PSH THEN B DCR =0 IF 1 HL LXI
```

You're done, but in 16K RAM this slightly larger code will overcrowd the SORT demo program on earlier copies of V1.9. Avoid the crunch by having SORT temporarily FORGET SCR, just ahead of : TASK ; on Block 45 Line 0. And put back the editor later, by ending Block 44 Line 15 with:  
 FORGET TASK 22 3 HEX LOADS DIR .

**IF YOU ARE NOT UP TO LOWER CASE:**

Well, not everyone has it yet! MMS recommends that you upgrade your TRS-80 Model I to support lower-case characters, preferably via Radio Shack's own upper-lower case hardware modification to the keyboard unit. Commencing with our Version 1.9 this mod is supported with a upper-lower case driver routine, written in Forth and incorporated into the precompiled MMSFORTH that comes up upon booting the system.

But if for reasons of your own your hardware doesn't support our lower-case driver, you have several options. You can keep it aboard and attempt to ignore the occasional "garbage" characters which show up instead of lower-case text, or you can use the ALL-CAPS routine included on the UTILITIES DISKETTE to rework the text on the diskette before attempting to work with it in RAM.

Perhaps the simplest and most overlooked approach to this problem is to recompile the Forth system, after removing the command which loads its lower-case driver block. This is Block 66, and it is invoked from Block 18 Line 14. (Block 67's full-ASCII keyboard driver is invoked simultaneously, and can be kept or removed at this time.) Change the 66 2 LOADS to 67 LOAD, or delete it completely to make both routines off limits. Also be sure to add the following new definition of CRT, ahead of PRINT. (The old one was on Block 66.) On the same Block 18 is a good place to do it:  
 4BDO CONSTANT CCRT : CRT CCRT OUT-UNIT ! ;  
 Then recompile with:  
 HEX FORGET OCTAL OC OD LOADS DIR CUSTOMIZE  
 to create a modified MMSFORTH System which sees all lower-case letters as their upper-case counterparts.

**FUN & GAMES**

**LIFE PATTERN:**

Try this square and unusually long-lived LIFE pattern. Begin by Moving down to X=24,Y=30. Then Draw 6 to the right, 12 up, 12 to the left, 12 down, and 6 to the right. Press G to generate the LIFE pattern!

**THE LAST WORD:** "Your FORTH language is Maravilhosa!"  
 - Edgar Pullen, a user in Brazil

