GENERAL DESCRIPTION OF MMSFORTH

MMSFORTH - a World of Difference!

For Radio Shack TRS-80 Models 1, 3 and 4,
IBM Personal Computer and Compaq

We're serious. As you will discover as you read through this information, MMSFORTH **is** different, and is a whole new software world - **a total software environment** - rather than just one more program, language, or operating system. So we have a lot to say. But first, since being different means adjusting to some new ways, let's briefly consider some of the potential advantages.

WHAT IS FORTH?

FORTH IS SPECIAL. It's **a living computer language** - one which adapts your choice of new words into its instruction set for the ultimate in modular, structured and personalized programming. Forth is interactive like an interpreter, fast like a compiler, very compact, and it even implements virtual memory. More and more programmers are discovering it to be ideal for the development of powerful computer applications which significantly stretch the limits of the available hardware.

WHY MMSFORTH?

MMSFORTH IS THE SPECIAL FORTH. It's the hot one with all the applications, plus plenty of strong utilities and professional support. Not another figFORTH, MMSFORTH is custom tailored to several popular computers and does not depend on an inefficient DOS to bottleneck its finest characteristics. Like most other up-to-date Forths, MMSFORTH includes the **79-STANDARD** nucleus of Forth words; it also offers a **greatly expanded vocabulary** with smooth implementation and great efficiency of action.

WE HAVE BEEN AROUND FOR A WHILE. Since early 1979 many thousands of persons and organizations have licensed MMSFORTH, making it the **most popular** professional Forth system. MMSFORTH offers **significant advantages** over any other Forth currently available for the TRS-80 or IBM Personal Computer, and many features supplied by MMS are not available in other versions of Forth for any microcomputer.

MMSFORTH™

A VERY BRIEF HISTORY. Invented in 1970 by a single person, Charles Moore, to simplify his own programming, the Forth language and system quickly became the computer environment of choice at many of the world's foremost radio telescope observatories. Excelling at first at process control tasks such as telescope motion and photography, it has grown and matured and proved equally revolutionary and practical for microcomputer business programs, games and I/O device controls. Now that the investment for a professional version is reduced from many thousands of dollars to as little as $130, thousands of new Forth users and programmers are enjoying its benefits.

WHAT IS FORTH LIKE? Mostly itself! Some compare it to Pascal, also a structured language, but **in Forth the programmer easily defines additional commands as they are needed.** The commands can be immediately compiled into the language's instruction set, and deleted when no longer needed. This makes Forth expandable and contractible so you can tailor it to your own needs. Yes, this means you can define any instructions the hardware can do, in your own choice of word names. No need to pre-load or "poke in" assembler subroutines as one normally does in BASIC, either, because assembler code can be inserted directly into a Forth program using a function of the language itself.

STACKS AND RPN. Forth is a stack-oriented language. Like Hewlett-Packard calculators, Forth uses Reverse Polish Notation. Unlike H-P's, Forth has **multiple stacks** of "infinite" depth limited only by the size of available memory. Forth programs tend to be so **compact** that a 32K disk system becomes practical for small applications. (16K special projects are possible.)

IT'S FAST! Persons accustomed to BASIC interpreters won't believe the **execution speed** of Forth programs. A demonstration program in the MMSFORTH package sorts 1,000 items in just 3 seconds on TRS-80, 1 second on IBM PC. MMSFORTH runs significantly faster than most other Forths, and approximately 20 times faster than equivalent interpretive BASIC programs (faster still when using Forth's Assembler). Forth's **modular, structured code** makes development time exceptionally brief and bug-free. All functions, even assembler code, can be tested interpretively during development of a package. We provide machine code dump routines and a search utility. Incredibly, Forth supports programming in high-level Forth, assembler, and machine code as well as functioning as an **interpreter and compiler**, all concurrently!

MMSFORTH INCLUDES UTILITIES FOR EASY PROGRAMMING

* the finest full-screen Editor you're apt to find
* full inline Assembler (8080 for TRS-80, 8088 for IBM PC)
* auto repeat keyboard with complete ASCII set, variable-character blinking cursor
* User callable Disk I/O with verification, and Tape I/O routines
* hooks to incorporate your own I/O routines for special devices
* virtual memory with multiple block buffers
* format or backup any range of tracks
* Copies utility to copy between **differing disk formats** (including standard TRS-80 and IBM formats and our own 5K/track format for 195K, 395K or 795K/drive with 40-tr.SS, 40-tr.DS, and 80-tr.DS drives, respectively)
* **user-definable windows** (IBM PC only)
* low-resolution graphics; on IBM PC, can run **16 colors or monochrome** display
* double-precision integer math (to +/-21,000,000.00)
* Extended-BASIC-like string-handling

MAXIMUM FLEXIBILITY. Because MMSFORTH runs in **its own software environment** without DOS, it is free to achieve an unusual degree of flexibility. Its disk format frees 1.5 tracks **more data space** than most DOS diskettes (plus an optional 20Kbytes additional data per "M.3"-format disk on the IBM), runs faster, adjusts for up to 8 35-track to 255-track drives, and permits MMSFORTH program and data diskettes to be read, changed, and run **interchangeably** between TRS-80 Model 1 and Model 3 or 4, or between Model 3/4 and IBM Personal Computer hardware. Just put the diskette in the other type of computer, change disk format on any drive, and keep running!

FLEXIBILITY FOR THE PROGRAMMER. Source code is provided for most applications and most of the language. (The lower portion of MMSFORTH is modified via system constants tables, for greater system reliability and portability.) Your complete professional application systems also can be precompiled with each final user's individual serial-numbered MMSFORTH System. You may choose to deliver to the final user precompiled without your source code; such programs will load rapidly and run efficiently, while effectively preventing user review and modification. MMS encourages professional programmers to modify and extend its system in this manner, provided that the necessary MMSFORTH System and any utilized accessory modules are properly sold to each final user. Bulk licensing of stripped, run-time systems is also available.

FREE PROGRAMS, TOO: Instructive, useful, enjoyable. All significant additional reasons to value the MMSFORTH System, all with full source code, and all free with your initial purchase!

* a Number Guessing game
* a Name Sorting program
* a remarkable **visual** demonstration of five common sorting routines
* a Graphics editor
* the Game of Life
* a menu-driven **Checkbook Balancing** program
* the famous **BREAKFORTH** arcade game with sound
* THE NOTEPAD **letter writing** program
* on IBM, a Kaleidoscope graphics demo that runs on the monochrome as well as the color monitor
* on IBM, a version of the Simon game with sound
* a fast and flexible **SEARCH utility** pops you into the Editor at each place to change, or prints them all

An optional directory block has been implemented for these demonstrations and is available for user applications.

NEARLY EVERYBODY CAN USE IT. In the early 1970's, Forth was a language for the programmer's programmer - a powerful but rough-hewn tool used by a knowing few to whittle out marvelous solutions to special problems at the radio-astronomy laboratories in which it grew. But now that Forth has evolved into a smooth system with good documentation, it is also liked and used by more and more hobbyists, professional program developers, and thousands of users of the MMSFORTH series of applications programs.

IT'S A GOOD TIME FOR YOU TO MEET MMSFORTH! It's the hottest new software environment for the IBM Personal Computer, Compaq, TRS-80 Models 1, 3 and 4, and our MMSFORTH Users Manual of over 200 pages makes learning it easier than ever before. This manual is available separately (without Appendices, but with a fine introduction and even a detailed analysis of our Checkbook Balancing program), for those who wish to evaluate Forth further or to use it with a version other than MMSFORTH.

MMSFORTH ADD-ON APPLICATIONS, UTILITIES AND SUPPORT

A WEALTH OF HIGH-PERFORMANCE MMSFORTH APPLICATION PROGRAMS: THE DATAHANDLER streamlined database manager, DATAHANDLER-PLUS expanded database manager for IBM PC, FORTHWRITE wordprocessor, FORTHCOM communications module, GENERAL LEDGER for accounting, TRADESHOW for high-speed commodities information, GAMES for everyone, and always more in development. They feature **powerful capabilities, surprising speed,** and **ease of use.** You don't need to know MMSFORTH to use them, but they come with **source code** for those who do.

WE ARE ALWAYS IMPROVING. MMS continuously adds refinements, additional modules, and application programs to its MMSFORTH system. Licensed users are invited to participate in our beta-test program for draft upgrades.

UTILITIES DISKETTE

The MMSFORTH **UTILITIES** Diskette includes:

* a very full FLOATING POINT MATH utility (computer must have BASIC in ROM) with **complex numbers, radian and degree trigonometry, rectangular and polar coordinate conversions,** etc.
* XREF, a powerful **cross-referencing utility** which quickly lists the anatomy of any Forth program's source code.

TRS-80 version only:

* a full Z80 ASSEMBLER utilizing most of MMSFORTH's capabilities for conditional control structures, interpretive and compiled operating modes, etc.

IBM version only:

* the TGRAPH color graphics wordset, including a full **Turtle Graphics** implementation with **graphics screen dump** to certain printers

New for IBM! V2.2 introduces:

* a very fast and high-precison **8087 version** of the FLOATING-POINT wordset
* rapid **n-precision integer math** routines with a full set of words. It can do such jobs as 100 factorial with **no** loss of precision!

FORTHWRITE

FORTHWRITE is our fast, full-performance word-processor with easy-to-learn simple operations and surprisingly powerful advanced capabilities. Designed for serious business, educational and professional applications, its list of features is awesome. For example, you can:

* do instant, screen-oriented text editing in windows from 1 to 255 characters wide
* move the cursor or delete to next or prior character, word, sentence (or control code), paragraph, page or block
* easily set tab stops (one at a time, by uniform increments, etc.)
* center, flush-left, flush-right, justify both margins, even force several items such as name, date, and page number to justify neatly across a single line
* find and replace quickly and with flexible options (including control characters)
* define multi-line headers and footers with flexible page numbering
* manipulate text blocks
* set absolute and/or **relative** margins by paper position
* indent new paragraphs, or **outdent** them (like this)
* use FORTHWRITE's powerful Include function to create efficient, nesting block, document (file) and keyboard-input linkages anywhere in text (ideal for "boiler-plate" paragraphs, and permitting insertion of data into forms, full-disk print-out, and keyboard entry of custom items during print-out)
* insert our print formatting commands or your special printer ASCII-code sequences anywhere in the document
* select standard parallel and serial printer drivers
* select special printer drivers to utilize extra features of many popular printers: toggle between several constant pitches with proportional spaces (between words) or full proportional type fonts, toggle subscripts, superscripts, underlining, dash-out, bold printing, etc.
* print **full-proportional** text on appropriate printers **with tabbing!**
* print from disk or active memory, or dump the screen at any time
* use the **print-to-disk** option
* generate form-letters, labels et al from DATAHANDLER or DATAHANDLER-PLUS file records
* do list processing **without** need for other programs
* maintain multiple, automatically **alphabetized directories** of FORTHWRITE documents
* simultaneously access multiple directories
* use your cassette tape recorder for keyboard-controlled transcription

Many other FORTHWRITE features are so advanced that their power may be lost on beginning users. Here are a few hints. On replacing a phrase, FORTHWRITE automatically checks and capitalizes the first character if it is at the beginning of a sentence! Or can search and replace using wildcard characters. When replacing multiple wildcard characters, it even "remembers" each original character in the wildcard positions!

FORTHWRITE sub-documents are of limited size (typically, 3,000 to 7,000 English words in 48K or 64K RAM) for maximum speed of operation and for user comprehension. One can **edit and print composite documents** as large as the multi-disk capacity, using the alphabetical organization of the FORTHWRITE directory or with Include functions.

Because FORTHWRITE runs in MMSFORTH instead of any DOS, it can even **use TRS-80 data disks in an IBM PC!** It comes with an XFER Utility to transfer text from Forth blocks, IBMDOS or TRSDOS. For example, this actual copy was transferred from a SCRIPSIT/TRSDOS file using XFER, then was modified and printed using FORTHWRITE with an NEC Spinwriter.

While FORTHWRITE's list of features is impressive, more important for many professional users are the **ease and speed** with which it performs. You actually have to learn relatively little to start being productive. FORTHWRITE smoothes user training and retention with very logical mnemonics for its keyboard controls, plus on-screen prompts, help screens, sample documents on disk, and a good users manual (complete with production-oriented sample documents for letterform, lease agreement and a will). Unlike many other word processors, its Insert mode works directly within the text, letter by letter, while you type and the display continues to wordwrap **without hesitation.** It can search for or move text blocks at the phenomenal rate of 10,000 characters per second (IBM PC), and can provide equally productive advantages in screen formatting, replace, and most other functions. Our users commonly applaud these features for removing the computer bottlenecks which previously obstructed their creative momentum.

DATAHANDLER

THE DATAHANDLER, a favorite and inexpensive first application for most new users, is an outstandingly fast, flexible, and **easy-to-use interactive data base** management system. It is based on SCELBI's PIMS Manual, but good programming in MMSFORTH dramatically increases its capabilities, storage capacity, and operating speed. With hundreds of entries, **find any record conditionally on any field or fragment thereof** with delay time less than a half-second; or sort the same files on multiple fields in 5 to 10 seconds.

Most data file projects for personal use will fit into THE DATAHANDER easily, as will many professional tasks. Efficient for files or sub-files of hundreds of records, it is an outstanding solution to a wide variety of personal and professional tasks. We use it for management of alphabetic and numeric data files such as inventory, mail lists, check register, collections, any time we have to retrieve on various different criteria.

DATAHANDLER and its sequential data files are **remarkably compact.** An indexed key structure incorporates sophisticated string and value selection mechanisms including normal compares and values inside or outside a range. If you use a TRS-80 or require access to source code, THE DATAHANDLER is **the best professional solution** for those data files which can reside in RAM: typically at one time, 24K in a TRS-80 or 34K in an IBM PC. Often, larger files can be divided into a series of subfiles for effective use. The user can specify up to 10 data fields appropriate to each particular job (or, with a minor change, up to 255), of variable length up to 255 characters each! Standard and special report formats can be output to screen and printer. DATAHANDLER files may also be used by DATAHANDLER-PLUS, FORTHWRITE and FORTHCOM.

The flexibility of its interactive program makes THE DATAHANDLER the logical choice for non-programmers. For the professional programmer, it and MMSFORTH offer far greater execution speed than BASIC programs, while its complete source code and Forth's structured and modular language provide the tools for rapid and accurate modification to a wide variety of custom applications.

Several MMSFORTH features make THE DATAHANDLER unusually well suited to single disk drive use: the program area of the DATAHANDLER diskette is **software write-protected** while the data file area is left open; alternatively, after loading the program its diskette may be replaced by one or more diskettes containing data only. Regularly used system configurations may be precompiled for five-second loading time.

Like other MMSFORTH software, THE DATAHANDLER comes with demonstration programs, data files, documentation, and full source code for further modification. Included on The DATAHANDLER diskette are mail list, checking account and bibliography applications with custom report commands and sample data files, both as useful products and as informative examples of other practical applications.

DATAHANDLER-PLUS

If you like DATAHANDLER, you'll love **DATAHANDLER-PLUS!** This major redesign and escalation of THE DATAHANDLER requires an IBM PC with a minimum of 128K RAM and does not include source code. DATAHANDLER-PLUS is ready to take your DATAHANDLER files and modify them to its expanded capabilities. (Single-density files cannot be read directly by IBM PC hardware.) You will profit from its up to 640K file buffer area, advanced keyboard command features building on those of FORTHWRITE, and **easily "trained" function keys.** You easily "preset" your own "views" in either user-definable single-record screen (form) or "**spreadsheet-type**" (table) format. Preset views, sorts, selections, print-outs, etc., are easy to define, modify, and retrieve as a part of your file. Best of all is the incredible DATAHANDLER-PLUS "**active window**" -- a word-processing window you use directly in any field of any record in these form or table views.

TRADESHOW

TRADESHOW converts an inexpensive TRS-80 Model 3 or 4 to a specialized high-speed intelligent-terminal for the selective remote monitoring and recording of up to 100 types of **Commodities Exchange transactions** from the COMEX or similar data lines (separately leased services) at **1200 baud** without modification. TRADESHOW is designed for easy and effective use by sophisticated commodities traders who value its selectivity and minimum **15-minute leadtime** on other data services.

FORTHCOM

With the **FORTHCOM** communications program in MMSFORTH you can perform a wide variety of RS-232 and **telephone communication** activities. Run your computer as a remote terminal on other computers, transfer "non-compatible" remote files with Forth files (DATAHANDLER, DATAHANDLER-PLUS and FORTHCOM) or Forth blocks (programs, data, NOTEPAD files) or your data back to DOS files (on **whatever computer**!). On TRS-80, spool the input to your printer. Between two FORTHCOM systems, transfer ranges of Forth files or blocks (with **space compression** and **error checking**) or even let one computer take over **remote operation** of the other. These capabilities are available through FORTHCOM's on-screen menu at up to 1200 baud, or directly to the programmer by modifying its Forth source code for special tasks at up to 9600 baud.

GAMES

The MMSFORTH **GAMES** Diskette includes a handy copy of the BREAKFORTH arcade game from the System Diskette, plus new arcade games (CRASHFORTH and FREEWAY), board games (OTHELLO and TIC-TAC-FORTH), and an excellent creation and solution aid for CRYPTOQUOTE puzzles. It's top-notch for fun and for learning interactive Forth techniques! All our games are provided in **high-level Forth source code,** and are so fast that adjustable speed settings are included to make learning possible.

GENERAL LEDGER

The MMSFORTH **GENERAL LEDGER** was contracted by professional accountants who recognized the short-comings of commonly available microcomputer packages and needed something better. It provides fast, smooth-operating accounting support including subsidiary ledgers and journals. It has an unusually **large capacity** of 500 accounts (using your choice of 4-digit account numbers) with unlimited subsidiary ledgers and over 2,500 line items per session (one group of journal entries). Its many built-in **information safeguards** include error checking, a complete audit trail, 60-character transaction descriptions, and prompting for additional data on unbalanced entries. Our GENERAL LEDGER's **optimized user interface** may double your data entry speed by minimizing switching back and forth from numeric keypad to other keys. Journal detail is always retained, so detailed journal and ledger listings may be made of any account or group of accounts (determined by any part of account number). Income statements include percentage information for present period and for year-to-date. Of course we include all standard accounting reports: trial balance, balance sheet, incomes statement (with percentages), journal listings, and account activity for any number of accounts.

MMS USER SUPPORT

MMS SUPPORTING SERVICES. Of course you'll have questions. MMS is well known for its good support of licensed users:

* good manuals and sample programs/files
* telephone tips
* inexpensive software updates
* User Groups worldwide
* the MMSFORTH Newsletter
* Forth-related books
* Forth workshops
* CUSTOM professional consulting
* system selection and hardware sales, too.

PRICE AND LICENSING ARE GEARED TO PROFITABLE USE. What **does** your computer time cost? A realistic analysis will reveal the worth to you of more efficient and more enjoyable tools. Several levels of MMSFORTH licensing offer cost-effective pricing for professional sites, and low prices for individual users, initial evaluation, or bulk distribution with **your** programs.

MMS IS A GROUP OF PROFESSIONAL PROGRAMMERS

SUPPORTING THE TRS-80 & IBM PC

HERE ARE TWO EASY WAYS WE CAN HELP YOU!

MMS CONSULTING SERVICES: Pay by the hour for the specific information or support you need. We can develop a business package, patch an existing program, or advise on hardware or software needs. Our system selection services can save hundreds of dollars and months of delivery time. We are as near as your telephone, and serve beginning hobbyists, small businesses and professional programming staffs.

MMS PRODUCT SALES: Buy selected hardware, prepackaged software and literature from a pro. It costs no more to get much of the best standard and special TRS-80 or IBM PC software from MMS. Don't expect to get everything. Do expect a careful selection of good capabilities and values in many areas of interest. Our disk drives, RAM and 8087 chips, printers and modems are tested here before shipping.

MMS HAS:

* worked (and played) with the TRS-80 and IBM PC since they were released.
* led TRS-80 training seminars for Radio Shack's own staff.
* developed MMSFORTH, the professional-level Forth system for TRS-80 and IBM Personal Computers.
* combined MMSFORTH and NEWDOS+ into the first 10-Megabyte hard-disk system for TRS-80 (in cooperation with Corvus Systems, Inc.).
* supported MMSFORTH with a growing series of major programs for general and specific use.
* provided its advanced MMSFORTH technology to companies developing and improving their own proprietary products.

Please contact MMS concerning your specific interests, preferably by telephone for prompt and accurate response.

MMS
MILLER MICROCOMPUTER SERVICES
61 LAKE SHORE ROAD, NATICK, MASS. 01760
(617) 653-6136

First impressions on this language.

MMSFORTH

MMSFORTH
Miller Microcomputer Services
Natick, MA
$89.95 cassette (V1.8 for M.I)
$129.95 disk (TRS-80)

Nicholas Spies
434 Grace Street
Pittsburgh, PA 15211

I was intrigued for some time by ads in *80 Microcomputing* for "MMSFORTH, The Professional Forth for TRS-80" (Miller Microcomputer Services). I read with interest claims that MMSFORTH was 10 to 20 times faster than Basic, that it was stack-oriented, that you could add your own commands, that it had a great editor, a variety of utilities, and that MMS would provide professional support. It seemed almost too good to be true.

I read some articles on Forth which further whetted my curiosity (A First Look at Forth, *80 Microcomputing*, July 1981). But even with program examples, the structure and syntax of Forth make it difficult for the newcomer to appreciate its many advantages over Basic.

Basic's Shortcomings

Basic was developed at Dartmouth College in the mid-60's to take advantage of the miracle of time-sharing. Suddenly computing power was available to anyone with a computer terminal and dimes for the phone bill. This caused a minor revolution; the power of the fabled mainframe computers was unleashed on nearly every college campus and business.

Basic fulfilled the need for an easily understood language of students and non-professional users. It was easily understood, could be written interactively, was fairly standardized (at least at first), and ran fast enough for a 110-baud terminal. Basic was derived from Fortran, originally designed to run batch programs by loading IBM cards into the mainframes of yesteryear. This may explain why Basic seems so stodgy today.

Why, in this day of CRT displays and floppy disks, do we still number all lines, edit by line rather than by screen, assign variables codes rather than names, call subroutines by line number instead of by name, struggle to get system-level access, contend with a *mysterious* disk system, and shift continually between Basic and DOS modes.

Yet Basic survives and probably will continue to thrive for some time; it is burned into the ROMs of every TRS-80 Model I and III, Color Computer, Pocket Computer, Apple, PET, and so on. But you can bypass the ROMs by loading a completely different language into your TRS-80 and enjoy programming from changed perspective. I used MMSFORTH; now my Model I TRS-80 has a new personality.

What follows is not a comprehensive product review but a subjective appraisal by a first-time user. Many functions of great interest are not covered. They are beyond the scope of an introductory article or I do not even know about them . . . yet.

MMSFORTH—First Impressions

MMSFORTH is available for the TRS-80 Models I and III on disk or cassette. The Model I and cassette require only 16K to run; the Model III requires 32K. Both disk systems need only one drive.

I ordered MMSFORTH by phone; it took about 10 days by first class mail. The System, Program and Utility disks arrived in an attractive three-ring binder with a 126-page instruction manual. The manual includes chapters on Forth operations, editing commands, input/output to disk and printer, data declarations, text handling, conditionals, branches and loops. Three programs, ranging from easy to difficult, provide detailed study notes. This is supplemented with instructions for using MMSFORTH system utilities and a system index.

The optional Utilities disk includes a Cross-Reference utility (XREF), Floating Point Math, and a Z80 Assembler.

For the rest of the documentation the user must sign two license agreement forms promising not to sell or give away MMSFORTH. Programs written for sale require the end user to have the MMSFORTH system owners or be under a separate licensing agreement. Personal use and backups are not restricted.

This seems to exclude MMSFORTH as the vehicle for commercial program development, unless the programs sell for far more than the System (or System and Utilities) cost. I hesitate to comment further without knowing what royalties are involved with the separate licensing agreement. Both disks have embedded serial numbers to deter the rip-off artist. I sent off the license forms and within a week had the rest of the documentation (a complete glossary of commands, various system addresses, a memory map and more material on the Assembler).

I also got an MMSFORTH Version 2.0 Modification Advisory, cures for bugs in their new system. The changes were easy to do using the editor. Newer versions do not require corrections.

The documentation is complete and well written, but it is not a textbook on Forth programming and requires careful reading for the first-time user. (MMS offers other books and magazine articles on Forth, as well as their own "MMSFORTH Newsletter.")

Starting Up

First, Backup all the original MMS disks; Format and Backup utilities are provided. Make several backups to minimize use of your MMSFORTH masters.

The Model I System disk comes ready to run on a 16K system, but you can configure the disk to your particular hardware.

Customize permits you to specify the directory block, the lowest unprotected block, memory size (to protect a printer driver in high memory), printer margin (set left margin of listings), the number of block buffers, disk startup speed (delay before reading and writing after motor is turned on), and number of disk drives. For each drive you may specify single or double density (on Model III), number of tracks and track access speed (allowing a mix of different densities and drives on line). Finally, there is an Auto command to perform a sequence of commands on booting the

"All is written back to the disk and automatically configures the system whenever that disk is booted."

system. These parameters are written back to the disk and automatically configures the system whenever that disk is booted.

When you are more confident with the MMSFORTH system, you can further modify the system. You can select the maximum number of block buffers (from the default of two), editor-like functions while inputting from the keyboard (recall the last line typed, for example), a lowercase driver and a special printer driver for the MX-80.

Other utilities on the System disk allow copying a range of blocks to another location, finding any word in a range of blocks (with options to print and edit matches), and translating source code from Version 1.9 to the present 2.0 Version of MMSFORTH.

You can design the system to match your needs and hardware. Disk systems for Models I and III have cassette functions. With the Model III version it is possible to select single or double density disks. Model I and Model III users can exchange disks.

System Extensions

As delivered, MMSFORTH is limited to character (8 bit) and single-precision (16 bit) integer arithmetic. What this lacks in handiness and dynamic range is compensated for by speed. You can add the following from the System disk (depending on memory): double-precision integers, arrays, strings, random number, graphics, screenprint (works from any mode, and prints graphics on MX-80), cassette functions, clock, and a few other utilities. Loading them all takes an additional 8K when compiled onto a System disk using Customize.

Several demos and games on the Programs disk allow the novice to appreciate the power of Forth. Then there is a very impressive string-sorting demo, the game of Life (with a Doodling program to input patterns), another version of Life with an assembler core (about one generation per second), and Breakforth (a challenging version of Breakout with sound written entirely in MMSFORTH).

The optional Utilities disk has single and double precision floating point math derived largely from ROM routines to conserve memory. You can select radians or degrees modes for trigonometric functions, rectangular to polar coordinate conversions, complex numbers, imaginary numbers, and a superfast program to solve quadratic equations. A flexible cross-referencing program (XREF) will print (to printer or screen) all references to words you choose in the range of blocks specified. A complete Z80 Assembler rounds out the Utilities disk.

All the programs, extensions and utilities are written in MMSFORTH, not machine language. Customize them by editing the appropriate block of source code, or only those portions of a utility which apply to your program.

Blocks

The basic storage unit of the Forth language is a Block, made up of 1,024 characters (1K). Because a block fits perfectly on the TRS-80 screen, writing, editing and listing programs is a breeze. Just POP from block to block without scrolling. Three blocks fit nicely on 8 1/2 by 11 paper for listings. Each disk on the Model I contains 87 consecutive blocks numbered from 0 to 86. The Model III has 179 blocks per disk. Each block is comprised of four 256-byte sectors. Blocks are numbered consecutively from drive 0 to drive 3; the system behaves as if they were on one large drive. A relative addressing convention allows access to the Nth block on drive D in addition to absolute addressing.

Information is transferred in one-block units from disk to block buffers in RAM. Access to disk blocks is almost as easy as accessing RAM (either directly or under program control). With four drives you have the equivalent of more than 250K of virtual memory and more than 500K on the Model III. Files are created within blocks by calculating the offset from the first byte of the block to where a particular record is written within the block. Any format developed by the programmer can be used. A completely documented example of data file handling is included with a Checkbook program.

Words

Forth consists of a dictionary of words. Words can be up to 31 significant characters (without spaces); they act like named subroutines.

Words are defined as in this example:

```
: PAUSE 2000 0 DO LOOP ;
```

Here the word Pause is defined as an empty loop counting from 0 to 1999. The colon begins the definition and the semi-colon ends it. Once a word is defined it becomes part of the system until you turn the computer off or tell the system to forget it.

A word can be used to build other words:

```
: FLASH 15360 1024 191 FILL PAUSE 15360 1024
BLANK PAUSE ;
```

Flash whites out the screen (fills from memory location 15360 for 1K with graphic code 191) executes Pause (waits), blanks out the screen (fills the screen with ASCII 32), and waits again. The Fill and Blank functions are executed at machine-code speed; a timing loop prevents Flash from just being a blink. Define your own lexicon of functions and build your own language. (Use Customize to make Flash part of your own MMSFORTH version ready every time you boot up.)

Execute Mode

The Execute mode performs direct functions and enters temporary word definitions.

In the execute mode, if you entered the definition above for Pause you could run it right away, although it would not do much by itself.

After defining Flash you could still run Pause. This is clearly far more useful than the Basic immediate mode, limited to one line. You can define words until RAM is filled. To purge the dictionary from time to time, define a dummy word (by convention, Task) before defining your test words. The MMSFORTH words Forget Task remove Task and all words defined more recently from the dictionary.

You may redefine words; only the most recent definition of the word is active when it is executed. The system prints a message warning you of the duplication. The user has total disk access in execute mode. You can copy blocks, list blocks, get an index of a range of blocks, and load programs written in the edit mode for execution.

Edit Mode

To save programs, write them with the editor and save the source code to disk. When the source code loads from disk it is interpreted as if you were in the execute mode.

The editor is easy to use; I wish I had its range of commands in a word processor. In edit mode you can: insert or delete characters or lines; lock into insert mode; truncate to end of line; copy lines; treat the block as a continuous page of 1024 characters (that is, characters wrap around when inserting or deleting); and use the arrow keys to position the cursor anywhere on the screen.

When you want to quit edit mode, update the block and Quit. Your edit is stored in a block buffer, but not on disk. To force changes to disk for all Updated blocks, use the rather colorful word Flush.

To reenter edit mode, type E and the current block appears on the screen. You can review all blocks on disk with a shift Clear Down-Arrow (or Up-Arrow to go backwards). This visual disk search proceeds at better than one block per second.

The editor also can be used on non-Forth disks as a Superzap-type program with direct visual access to any byte on a disk. This can be useful for inspection of text files.

Assembler

It is easy to include 8080 assembler code in a program using the MMSFORTH word

"Edit and customize [utilities] by calling the appropriate block."

Code to start the sequence and either Next, PSH, or PSH2 to return (passing nothing, HL, or DE and HL to the user stack). The 8080 assembler is incomplete, but MMSFORTH words—like CMOVE (which moves a series of bytes from one memory location to another) render some assembler functions unnecessary. Control structures and word order for the assembler follow Forth conventions.

Data Types

Handle data in MMSFORTH by character, single, double and triple-precision integer, single and double-precision floating point and complex numbers.

In an 8-bit computer all data is stored at memory locations as 8-bit bytes. Each byte can represent 256 different bit-patters or states. These 256 states represent ASCII characters, graphics codes, unsigned (positive) numbers from 0 to 255, or signed numbers from minus 128 to plus 127 depending on the context. Numbers larger than 255 are represented by groups of two or more bytes called words (not to be confused with Forth words).

You can consider a 16-bit FORTH variable as two characters, as two 8-bit numbers or as a 16-bit number. The choice depends on the particular memory operator used to access the variable. Quite unlike Basic which is limited to the 8-bit PEEK and POKE commands, you can store and read from 8-bit characters to 64-bit complex numbers as units.

Strings are a maximum of 255 bytes long (the first byte is the length); string functions are a superset of Disk Basic string functions. Unlike Microsoft Basic there are no string pointers, making it easier to directly access and change strings in memory. $XCH (exchange strings) is useful in sorting programs.

Algebraic	RPN
(A * B)	A B *
((A * B) – C)	A B * C –
(A * (B – C))	A B C – *
(A + (B / (C – D)))	A B C D – / +
(A * (B / (C * D) + E))	A B C D * / E + *

Table 1. Algebraic vs. RPN

(A*(B/(C*D)+E))

Entry	3rd	2nd	1st	TOS
A				A
B			A	B
C		A	B	C
D	A	B	C	D
*		A	B	(C*D)
/			A	(B/(C*D))
E		A	(B/(C*D))	E
+			A	((B/(C*D))+E)
*				(A*((B/(C*D))+E))

Table 2. Stack values during execution

Memory Functions

The MMSFORTH words to locate numbers or strings in memory are:

'<word> "tick"—gets memory address of <word>, where <word> is the name of a variable, constant or system word
@ "fetch"—gets contents of memory location
! "store"—stores to memory location

```
BLOCK 104:

 0 ( DYNAMIC STACK DISPLAY  PAGE 1 OF 3  1 AUG 81 N. SPIES )
 1 ( MAKES STACK VISIBLE AND PERMITS FUNCTIONS )
 2 : TASK ;
 3 : SETDISPLAY PAGE   ( SET DISPLAY, LABELS, PROMPTS )
 4  0  0 PTC ." STACK DISPLAY"
 5  1  5 PTC ." TOS"   2  5 PTC ." 2ND"    3  5 PTC ." 3RD"
 6  4  5 PTC ." 4TH"   5  5 PTC ." 5TH"    6  5 PTC ." 6TH"
 7  7  5 PTC ." 7TH"   8  5 PTC ." 8TH"    9  5 PTC ." 9TH"
 8  1 37 PTC ." N  NUMBER"  6 37 PTC ." R  ROT" 11 37 PTC ." -"
 9  2 37 PTC ." D  DROP "   7 37 PTC ." <"      12 37 PTC ." *"
10  3 37 PTC ." U  DUP  "   8 37 PTC ." ="      13 37 PTC ." /"
11  4 37 PTC ." S  SWAP "   9 37 PTC ." >" 14 37 PTC ." M  /MOD"
12  5 37 PTC ." O  OVER "  10 37 PTC ." +" 15 37 PTC ." BREAK"
13               ( CAUSES NEXT BLOCK TO BE LOADED )  -->
14
15


BLOCK 105:

 0 ( DYNAMIC STACK DISPLAY  PAGE 2 OF 3  1 AUG 81 N. SPIES )
 1 : DSTACK 1 15 PTC  'S       DUP U. ? ."      "  ( TOS , @ )
 2          2 15 PTC  'S   2 + DUP U. ? ."      "  ( 2ND , @ )
 3          3 15 PTC  'S   4 + DUP U. ? ."      "  (   ETC   )
 4          4 15 PTC  'S   6 + DUP U. ? ."      "
 5          5 15 PTC  'S   8 + DUP U. ? ."      "
 6          6 15 PTC  'S  10 + DUP U. ? ."      "
 7          7 15 PTC  'S  12 + DUP U. ? ."      "
 8          8 15 PTC  'S  14 + DUP U. ? ."      "
 9          9 15 PTC  'S  16 + DUP U. ? ."      " ; ( 9TH TOS )
10 : POS    0 40 PTC ;       : POS1 0 23 PTC ; ( CURSOR POSITION )
11 : EL    30 EMIT   ; ( CLEAR TO END OF LINE )
12 : ENUMBER  POS1 EL ." NUMBER - ENTER" #IN POS1 EL ; ( NUMBER )
13 : EDROP      DROP     POS  ." DROP"     EL ; ( DROP )
14 : EDUP       DUP      POS  ." DUP"      EL ; ( DUP )
15 : ESWAP      SWAP     POS  ." SWAP"     EL ; ( SWAP )   -->


BLOCK 106:

 0 ( DYNAMIC STACK DISPLAY  PAGE 3 OF 3  1 AUG 81 N. SPIES )
 1 : EOVER  OVER POS  ." OVER"          EL ; ( OVER )
 2 : EROT   ROT  POS  ." ROT"           EL ; ( ROTATE )
 3 : E<     <    POS  ." LESS THAN"     EL ; ( LESS THAN )
 4 : E=     =    POS  ." EQUALS"        EL ; ( EQUALS )
 5 : E>     >    POS  ." GREATER THAN"  EL ; ( GREATER )
 6 : E+     +    POS  ." ADD"           EL ; ( ADD )
 7 : E-     -    POS  ." SUBTRACT"      EL ; ( SUBTRACT )
 8 : E*     *    POS  ." MULTIPLY"      EL ; ( MULTIPLY )
 9 : E/     /    POS  ." DIVIDE"        EL ; ( DIVIDE )
10 : E/M   /MOD  POS  ." /MOD"          EL ; ( /MOD )
11 : STACK SETDISPLAY   BEGIN   DSTACK   POS1 ( MAIN PROGRAM )
12   " SELECT BY KEY :"  KEY  ( WAIT FOR INPUT, BRANCH ON INPUT )
13  ACASE NDUSOR<=>+-*/M" ENUMBER EDROP EDUP ESWAP EOVER  EROT
14     E<   E=   E>   E+   E-   E*   E/   E/M  ( ROUTINES )
15  CASEND 0 UNTIL ;    STACK  ( EXECUTE PROGRAM WHEN LOADED )
```

Program Listing 1. Stack Memory Locations

"Words can be up to 31 significant characters... they act like named subroutines."

Use prefixes for memory, math, and stack operations applied to numbers other than single-precision integers. Speed and programming flexibility are gained.

Constants, variables and arrays can be defined for any data type, including strings. A constant's name evokes its value; a variable's name evokes its memory address making it easy to change its value with !. A variable's contents are fetched with @.

The Stack

You can write complex programs in Forth without one variable or constant. Forth uses a stack (a set of memory locations) to hold values and pass them from word to word. Passing numbers from one word to another is as simple as leaving the results of one calculation on the stack for the next word to pick up. The result is a great savings in coding.

The stack is just a pile of numbers in memory with a top-of-stack (TOS) pointer showing the top of the pile. As numbers are entered into the stack the pile gets deeper. The TOS pointer always indicates the most recently entered number. Numbers can only be entered and taken away from the TOS. This sort of stack is also called a Last-In First-Out (LIFO) stack.

All Forth functions involve the numbers in the TOS and the stack positions immediately below. When a function is executed, the stack pops up removing the parameters used in the function and making the stack less deep. The result is left in the TOS.

To use a LIFO stack with its semi-automatic management of number, you cannot use ordinary algebraic notation, with its parentheses and equals signs. Instead you use a more efficient notation called Reverse Polish Notation.

Reverse Polish Notation

RPN is a generalized way to evaluate formulas developed by J. Lukasewicz. RPN was named both in his honor and because it seems backwards to those used to algebraic notation.

Table 1 shows some comparisons between algebraic notation and RPN.

To interpret an algebraic expression in RPN, take the operands (ABCDE) from left to right until there is an intermediate result (close-parenthesis) where the appropriate operator (+ − * /) is inserted.

Each operand is pushed into the stack (further and further if there are no intermediate results) and then popped back up as operations are performed on the top two numbers in the stack. Looking at the last expression in Table 1, the stack would contain the values during its evaluation as shown in Table 2.

In MMSFORTH in execute mode, for A = 2 B = 20 C = 4 D = 5 E = 30, the entry for the above example would be:

```
2 20 4 5 * / 30 + * . <ENTER> 62
```

Each time a value is pushed into the stack, the stack gets deeper, and each time an operation is performed the stack pops back up with the result of that operation. The result is that the original algebraic construct is faithfully executed, although the operators and operands are entered in an entirely different order. In addition to requiring fewer keystrokes than algebraic notation, RPN is relatively easier to implement in machine language. This saves memory and increases speed.

A variety of functions move around, duplicate and delete values on the stack to use the same numbers again and again if required. In Forth, all functions involving numbers and operations are noted in RPN. Thus 17 88 ESET is the same as SET (88,17) in Basic (set a graphics character). Forth's difference of order looks strange to the Basic programmer.

Programming

Writing a program in Forth is unlike writing a program in Basic. Programming in Forth consists of defining a hierarchy of words, up to the word that executes the programming. You add words of your own definition to the dictionary. Each word's definition must be in terms of system words or words defined earlier in the source code. The stack most often holds data to be manipulated by various words, although named constants and variables are also available. The main difficulty in learning Forth is visualizing and keeping track of values in the stack. The reward is faster execution using less RAM.

As a program takes form you feel more and more power at your fingertips. Each word includes within itself more of the previously defined words and all the functions they imply. Tying together the program at the highest level is often the easiest part.

Debugging and testing can be done by revising the source code with the editor and running the updated program or by trying routines in the execute mode using PCRT (display to both printer and CRT) or the screen-print utility to record what you have written. If an error occurs during execution, the EEDIT function places the cursor directly in the block where the error was sensed, ready to edit! Needless to say, this makes debugging far simpler than with Basic.

Program Listings

Program Listings are easy to read because each word is defined by words defined earlier. Enter notes on program flows and the actions of words parenthetically.

I have included two MMSFORTH utility programs.

Program Listing 1 shows the top nine stack memory locations and their contents dynamically, with the option to perform a variety of stack functions. Do not enter the line numbers.

Program Listing 2 shows the keyboard scan memory locations and how their values change when you press various combinations of keys. This is useful for game planning and where branching is based on keys pressed.

Conclusion

MMSFORTH is a complete version of the Forth language (a superset of the Forth 79 Standard) and makes the TRS-80 a very

```
BLOCK 107:

 0 ( KEYBOARD SCAN DEMO - PAGE 1 OF 1  28 JUL 81   N. SPIES     )
 1 : TASK ; ( DUMMY WORD TO FORGET PROGRAM AFTER EXECUTION )
 2   ( PROGRAM - FIRST PART PRINTS LABELS FOR MEMORY LOCATIONS )
 3 : KEYS PAGE 0 25 PTC . " 14337:"  1 25 PTC . " 14338:"
 4              2 25 PTC . " 14340:"  3 25 PTC . " 14344:"
 5              4 25 PTC . " 14352:"  5 25 PTC . " 14368:"
 6              6 25 PTC . " 14400:"  7 25 PTC . " 14464:"
 7 ( START PROGRAM LOOP - FETCH/PRINT CHARACTER AT EACH MEM LOC )
 8  BEGIN  0 35 PTC 14337 C?  . "     " 1 35 PTC 14338 C?  . "     "
 9         2 35 PTC 14340 C?  . "     " 3 35 PTC 14344 C?  . "     "
10         4 35 PTC 14352 C?  . "     " 5 35 PTC 14368 C?  . "     "
11         6 35 PTC 14400 C?  . "     " 7 35 PTC 14464 C?  . "     "
12  0 UNTIL ;   ( END FOREVER LOOP - BREAK TO EXIT PROGRAM )
13
14 KEYS    ( EXECUTE PROGRAM WHEN LOADED )
15
```

Program Listing 2. Keyboard Scan Memory Locations

(Reprinted with permission of
80-U.S. Journal, copyright 1982.)

powerful tool for developing programs. Build a powerful set of subroutines and build on your previous work with an ease unknown to Basic.

Be prepared to spend some time learning the system, getting familiar with the stack and RPN, and mastering the data types. Your effort will be well spent.■

Reviews

MMSFORTH Version 2.0
Model I/III, 32K with disk
Miller Microcomputer Services
61 Lake Shore Road
Natick, MA 01760
(617) 653-6136
$129.95

Like a lot of other people I know, I was introduced to the FORTH language by the August, 1980 issue of *Byte* magazine. At that time, I was not very impressed with it. After all, who wanted that much control over their computer? I put it aside as 'just another computer language.'

Over time, however, I kept finding articles on FORTH and, much to my surprise, found myself reading every one and even rereading several. Eventually, I had to face the truth: FORTH had reached out and grabbed me. Just the thought of having that much power over my computer was thrilling.

Finally, I could take it no longer, I had to have FORTH on my Model I. With credit card in hand, I boldly called Miller Microcomputer Services and confessed my obsession. The nice lady who had answered the phone was understanding and, after taking my credit card number, assured me that a copy of MMSFORTH would be on its way to me as soon as possible.

Four days later, it arrived. (Luckily for me, it was a Friday. I don't think I would have been much good at work the next day!)

Unlike most people with a new toy, I actually read the documentation before playing, so I settled into my reading chair and went to it. On the table of contents page was written this heart breaking notice: "** These Appendices are provided upon return of your properly completed MMSFORTH License Agreement." Sure enough, what I would consider the most interesting Appendices were missing. There was still more than enough information in the manual for me to get started, but I had this feeling of incompleteness.

I immediately filled out the single user License Agreement, put a stamp on it, and walked down the block to the nearest post box and mailed it. (I received the missing Appendices the next Friday.)

The manual is one of the best third party pieces of documentation I have seen for the TRS-80. Similar to the TRSDOS manual, the first part of the manual is a tutorial of how to use the language, while the second part (the Appendices) is an excellent reference manual. The first thing the manual has the new MMSFORTHer do is back up the master disks. Smart thinking! With MMSFORTH, the programmer is *king* and what the programmer orders, the computer does — even if it means destroying the master disks! As it says in the manual: "IBM says 'THINK!', MMSFORTH says 'BACKUP!'."

Once the neophyte MMSFORTH programmer (me, in this case) has working copies of the master disks and the masters are safely tucked away, the manual proceeds to teach the potential MMSFORTHer how to "go FORTH." In a short time, I was doing fairly complicated programming with very little pain. And when the other Appendices arrived, I could begin programming in earnest.

The two supplied disks, labeled "SYSTEM" and "PROGRAM," are almost completely full. One of the Appendices (and not one of the missing ones) contains an index to their contents. The "SYSTEM" disk contains well over half of the source code for the MMSFORTH language. It is through the use of this source code that the MMSFORTHer can customize a version of MMSFORTH to the exact needs required. Do you have a serial printer and the Electric Pencil lowercase mod? Just select the right options on the option select block, and you can generate a new version of MMSFORTH which supports them both. Do you want to be able to type in a line before the computer is ready, like when the disks are being accessed? Again, just select the option and regenerate your MMSFORTH!

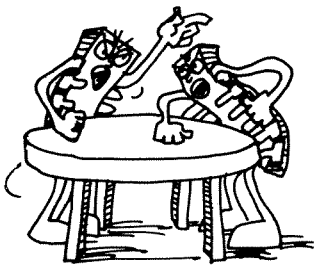
I should point out that this source contains an 8080 assembler, a full screen editor for source code, printer drivers, keyboard drivers, and interrupt routines — many things that a typical user wouldn't need, but a programmer quickly develops a need for.

The "PROGRAM" disk contains several examples and games in MMSFORTH (again, with the source code). Do you have a single disk system? Don't worry, MMSFORTH does not use overlays like TRSDOS does. Once MMSFORTH is loaded, you can take your system disk out of the drive and put a data disk in! (Warning: MMSFORTH, as it comes, does not read TRSDOS files. However, if you are clever, and knowledgeable about the TRSDOS file formats, you can transfer the data from TRSDOS to MMSFORTH.)

Also available from Miller Microcomputer Services are several utilities and packages written in MMSFORTH. One package is a personal database management system called "THE DATA-HANDLER," which I understand is very good, but I have not seen it. Another package is a "communications module" called "FORTH-COM," which I have seen and is excellent. (I use a slightly modified version of it to talk to an HP-3000 at 1200 baud without any problems.) Some of the utilities available are a full Z80 assembler, floating point numbers, and a word cross reference generator that can come in handy, especially during the testing of application systems.

If you have a TRS-80 Model I or III and have picked up an itch for FORTH, I have no qualms about recommending MMSFORTH to you. The manual is excellent, the language is excellent, and you will enjoy the feeling of power you have over your computer.

Anthony F. Pepin



Roundtable

The readers and editors present questions, answers, comments, and tirades.

Miller comments on Forth reviews

MMS has been following with interest the ***PC Age*** discussions on both the 8087 "fast-math chip" and, at long last, the Forth computer language we know and love. Apparently our own MMSFORTH System (V2.2) offers features missing from those you have so far reported; I would like to correct the impression that Forth, rather than some Forth products, is deficient in these regards.

The Forth introductory article by Brad McGehee (May ***PC Age***) accurately identifies Forth as a different way of seeing the computer and an outstanding one if approached in the right manner. The 8087 benchmark letter by Steven Ruzinsky in the same issue emphasized the point, when he nearly doubled the best prior run on the PC by using PolyForth in place of Pascal. But he casts some additional doubt on the usefulness of Forth when he says that he had to write trig function routines himself in order to run the test.

Please pass on the word that MMSFORTH has offered trig functions for many years. On the IBM PC, the MMSFORTH Utilities Diskette (a $39.95 option) provides a rich set of transcendental functions in radian and degree modes, complex numbers, rectangular to polar coordinate conversion, and much more.

The same Utilities Diskette supports the 8087 chip with these and QUAN-type single-precision, double-precision, and complex numbers, singly- and doubly-dimensioned ARRAYs of the above types, and so forth. Plus a cross-referencer, and a fast, accurate Turtle Graphics utility with screen-dump to printer. This is how Forth should be; evaluating it by using stripped versions is similar to evaluating MicroSoft Extended BASIC by trying Tiny BASIC.

Like Steve Ruzinsky, Tom Dowling of MMS wrote an 8087 Forth version of the FPBENCH test (by G. Scott Owen, in the March ***PC Age***). At least to a Forth programmer, I think this example shows that Forth code can be readable (see Figure 1). Tom's routine runs the 500 iterations in 2.8 seconds, which is 20% faster than Steve's and over twice as fast as the Pascal PC "winner" in Dr. Owen's article. By the way, the Pascal version took an improper shortcut by using SQR instead of the required $Y^{2.0}$; we can do that in 2.4 seconds, instead. But then, if the object was coding for maximum speed rather than an equivalent test, we would recode the innermost loop in MMSFORTH's full 8087 Floating-Point Assembler. It's on the MMSFORTH System Disk, as are our routine's TIME functions.

The common refrain that Forth's efficient RPN math is "not natural" can cause chuckles among Forth programmers, since RPN *is* natural to the computer and also to *you*. You were *retrained* to use Algebra. But, when you add or multiply a few two- or three-digit numbers on a piece of paper, you still write them down *before* doing the operation. (23 37 +) That's RPN.

Having established that MMSFORTH (and some other Forths) can help the programmer, let me also comment on Brad McGehee's belief that Forth won't substantially enrich the computer user's applications library. The reason MMS uses (and thus develops and markets) Forth is because of its applications power. We have offered strong applications, standard and custom, for years. Standard ones presently include THE DATA-HANDLER database system, FORTH-WRITE word processor, FORTHCOM communications module, GENERAL LEDGER in MMSFORTH, and our TRADE-SHOW smart terminal program for the high-speed Commodities Exchange data line. And, some very good GAMES.

```
Block 171 [171 :0]

 0 ( 05/25/83 8087 Benchmark, MMSFORTH V2.2 routine by T.Dowling )
 1 ( After articles in PC Age: March '83 p.56 and May '83 p.8. )
 2 9 FQARRAY X          FQUAN A   FQUAN Y
 3 : FPBENCH  ( -> )
 4   BEGIN    CR CR ." This is a number crunching speed test"
 5            CR ." Enter the number of iterations"   #IN
 6          0 0 0 SET-TIME ( initialize RTC timer )
 7     0 DO   PI  % 2.1  F/  IS Y
 8
 9          Y SIN  1 IS X   Y COS    2 IS X   Y % 2 X^Y     3 IS X
10          Y SQR  4 IS X   Y EXP    5 IS X   Y LOG         6 IS X
11          Y      7 IS X   Y PI F*  8 IS X   Y SIN % 2 F*  9 IS X
12
13          10 1  DO  I X  FDUP  F*  PI  F/  IS A   LOOP
14        LOOP  CR TIME ( display time ) ."  Finished, Again" Y/N
15   UNTIL  ;           FPBENCH

          Figure 1 - MMSFORTH Source Code for FPBENCH
```

They all deliver surprising speed and compactness, which translates into easily learned, and for the programmer they include structured, modular source code to facilitate custom modifications. I expect that our DATAHANDLER-PLUS, currently at beta test sites, will particularly delight old and new users.

As you now see, Forth can provide greater-than-reported benefits to the hot-shot programmer and the demanding user alike. We'll be happy to send literature on the MMSFORTH System to any reader. Thank you for helping to dispel some of the common misinformation concerning Forth.

For further information contact: Miller Microcomputer Services, 61 Lake Shore Road, Natick, MA 01760; (617) 653-6136.

A. Richard Miller

Valid points, Dick. Forth is one of the few, if not the only, remaining languages that exist on two levels. Forth is available as a public domain hobby product. The bare minimum is provided, a more experienced friend provides support, and the cost ranges from free to cheap. MMS and others offer a supported language with numerous and powerful extensions and utilities. The full function Forths are obviously in a different price range.

by A. Stewart Johnson

Get a Handle on Your Data

The advent of the microcomputer has opened the door to the use of computer technology by the small business professional. Selecting the right software, however, remains an obstacle to a system's full implementation.

The solution generally lies with reading the ads and reviews in the various journals and then hoping you make the right guess. If you are fortunate enough to be dealing with a reliable computer store or software house that provides support to the buyers of its products, your problems may well be reduced. Or you can pursue a third source, the computer consulting firm.

Could I computerize my mailing house?

My original interest in micros began because I am a radio ham who was looking for a modern means of adding teletype capability to my station. While reading the various computer journals, I realized there might be an answer to improving and streamlining my business procedures as the manager of a mailing house.

During several months of inquiring, I kept running into the name of Miller Microcomputer Services (MMS, 61 Lake Shore Road, Natick, MA 01760). I soon discovered that since 1977 it had been specializing in offering consulting services for owners of the TRS-80® Model I. I finally called and had a nice conversation with Dick Miller, the president. A couple of months passed. Finally, I purchased a TRS-80 Model I.

During this time I had specifically been looking for a good mailing list program. The market offered several such programs, but on closer examination none was of the professional quality I sought. You are probably asking, "But what is so complicated about a simple mailing list program?" A look at my requirements might serve to illustrate an important point that is often overlooked, namely that software is being written today by programmers who do not know their subject. Even though the software does its job, it is loaded with "bells and whistles" and is inefficient.

Automation and the adoption of uniform standards throughout the mailing industry are the reasons that so much business mail can be generated. This means, for example, that a mailing list generated for use on automatic addressing machines is generally formatted as four labels across on standard 14 7/8-inch paper, and the label measures 3.4-inch wide by 1-inch high.

My company, however, maintained lists with as few as 200 names on it, and our system generated only one-up labels. But, since customers with mailings running into the thousands were providing us with four-up labels, it was becoming increasingly expensive to switch our labeling machines back and forth between one-up and four-up labels. Enter the computer.

Those who are familiar with available mailing list programs realize most are not capable of meeting the four-up requirement. Additionally, if you format for four-line entries and you put in a three-line entry, a blank line will appear when the mailing label is generated. Similarly, there are other serious deficiencies in mailing list programs, such as the number of lines you can have and the number of characters per field.

Ideally, I needed a program that would generate at least one to four labels across on a 14 7/8-inch form and handle a mix of three, four and five-line addresses without printing blank lines on the mailing labels. The program should also allow up to 30 characters per each of five fields. Obviously, fast sort and string search routines were necessary. Would I find the required program?

(Incidentally, depending upon the printer used, a given number of labels can be printed considerably faster four-up then one-up.

For example, using a Centronics 102A printer, I can print 3,500 labels per hour formatted four-up as opposed to 900 per hour formatted one-up. The 102A printer is a dual-head, bi-directional unit rated at 330 characters per second (cps).)

The MMS Datahandler

A check of several nationally advertised programs, including a couple produced locally, provided nothing that would meet my needs, and queries about having them modified drew mixed reaction, suggesting that it would be a hassle and considerable extra expense with no guarantees. Where to turn?

I had been impressed by my conversation with Dick Miller and decided it would be worthwhile to make an appointment with him. It was time and money well spent. The visit introduced me to MMSFORTH, an extended version of standard *Forth*, and, more importantly, to The Datahandler, Miller's database management (DBM) system.

The Datahandler was written by Tom Dowling of MMS. A nontransferable single-use license for its use costs $59.95. The Datahandler, provided in MMS source code, requires the MMSFORTH Disk System, priced at $129.95 for the TRS-80 Models I and III, or $249.95 for the IBM Personal Computer.

Generally, most DBM systems require that a specific amount of space be reserved for each designated field, which means when data are entered for storage, even if all the reserved space is not used, the unused space still occupies space on the disk and, for that matter, in memory. No such situation exists with The Datahandler.

You merely specify the field and are automatically allowed up to 255 bytes in that field. Any space not used is not reserved in memory or on disk. Thus, it is possible to store more than a thousand names and addresses on a 40-track, single-density disk. I have used The Datahandler to store more than 1,500 three-line addresses on a 40-track, single-density floppy disk.

```
( 1 )
INSTIT   ,A: JACKSON,JOHN A.
DEPT     ,A: N.Y. SOC. OF CPA
NAME     ,A:
ADDR     ,A: 600 THIRD AVE.
CITYSTAT ,A: NEW YORK, NY
ZIP      ,A: 10016
CODE     ,A: CPA
```

Typical file record.

One specific trade-off, which is not necessarily serious, can be an annoyance. The Datahandler cannot do a disk sort. For my mailing lists, this means working in files containing from 250 to 350 records, depending upon the number of lines per address. This obstacle can be overcome if the user is willing to do a little pre-planning before entering the data.

If lack of a disk sort is an inconvenience, the in-memory sort routine is a major asset. Besides being fast (a look-up in less than a second), the program can do a multiple sort to all defined fields at once. For a mailing list, this means that you can do a complete and perfect Zip/Alpha or Alpha/Zip sort in one operation.

The Datahandler has a number of features that are common and routinely found in DBM systems. Their implementation, ease of use and the near impossibility of locking the computer and having to reboot with the attendant loss of data in memory, however, separate The Datahandler from other DBM systems.

Superior performance

The Datahandler's superior performance can be attributed to two key items. First, since it is written in MMSFORTH, the unusual efficiency and power of *Forth* allows you to run The Datahandler with 32K of memory and one disk drive. Compare this to any other decent DBM system in which a minimum of 48K and two disks is often required.

Second, it does not include a lot of "bells and whistles," such as numerous and sophisticated data input and output formats. It is quite possible for you to format and input into a Datahandler data file while someone else using one of the other programs is still trying to define an input format. This, of course, raises the question of selecting software and why a firm such as MMS represents a viable solution.

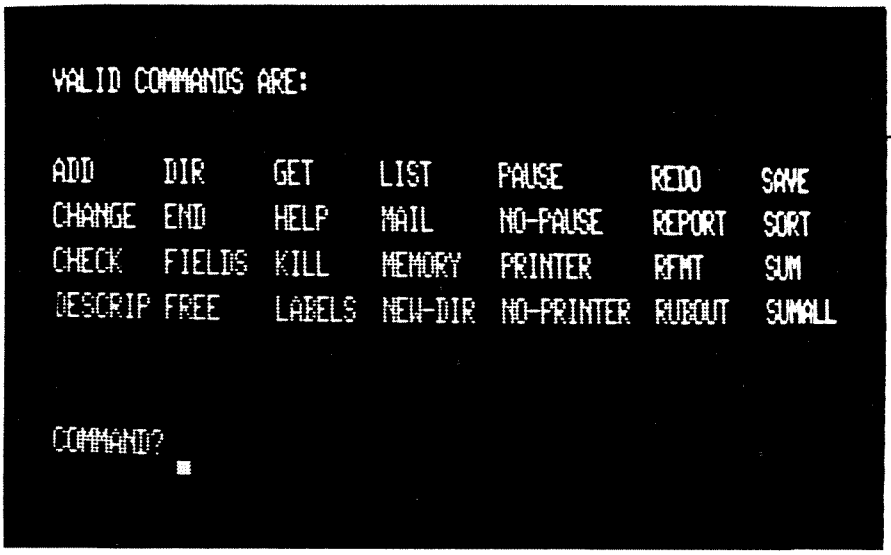
As noted above, the available canned software did not meet my requirements. First, software, like many products, is manufactured and marketed with the idea of appealing to as many prospective consumers as possible, hence "bells and whistles." Second, programmers often are trying to write programs on subjects in which their knowledge is limited. This is where a qualified consultant can bring order out of confusion.

Finding the right consultant is not the easiest task. A key to a successful search, though, is knowing what you want to accomplish and letting the prospective consultant propose a solution. In my case I had to depend on knowing my business thoroughly. If, on the other hand, I had been looking for a bookkeeping package, I would have consulted with my accountant. Eventual success will depend on the prospective user knowing his business and being able to communicate with the consultant

If you need a DBM system capable of all sorts of different formatted outputs, then The Datahandler would require extensive customizing. On the other hand, if your requirements are for a good solid data management system in which you essentially want to be able to store and retrieve data, then The Datahandler is ready to go.

The Datahandler in action

Upon booting the disk, you face the choice of bringing up an old file or setting up a new one. The standard program allows you to define up to ten fields, though more can be made available through a program change. (The first six fields



The "Help" command reveals the above screen.

are used for Mail, Datahandler's mailing list/report option.) The fields are optimized for sorting. In field #1, you can use the last name first, followed by a comma and the rest of the name. When printing, the comma will instruct the computer to print the first name first. For sorting ease, the Zip code gets field #6 for itself, but the mailing label routine will concatenate it to field #5.

The various search options offered cover about every combination you will need. For example, you can print all addresses within a particular state or all outside a particular state. Similarly, you can print all addresses within a given range of Zip codes or all outside that range. Using a checkbook program, which is part of the package, you can list all checks before or after a certain date. Value (numeric) and string searches are possible.

The Report format allows for variations sufficient for a professional-looking report. For example, names and addresses can be printed in a horizontal format, and the various elements can be in any order. Any or all fields can be selected, such as name and telephone number only, or inventory part number, part description, and quantity in stock. Allowing for the fact that a report might cover several pages, you can select page length and number of printed lines per page. For the standard paper length (11 inches), the program defaults to a page length of 66 lines with 60 printed lines. All of the various format parameters, though, are completely keyboard changeable. Here, the uniqueness of MMS and The Datahandler come into play as represented by my own experience.

Custom programming

The Datahandler, as provided, is designed to produce one-up mailing labels only, but I needed a four-up format. MMS was not only able to provide four-up capability, but, as a bonus, the ability to choose from one to five labels across. Additionally, I can specify a label width of from 20 to 50 characters.

I also needed the ability to merge files. No problem. Although I have to divide a disk into several files, I can output the entire disk to printer in one operation by a custom routine that automatically loads and outputs each file in order. The same routine tells me how many labels were printed.

The significance of this is that MMS and The Datahandler can give you the advantage of custom programming at a small fraction of the cost of such programming. In short, you can purchase The Datahandler as an inexpensive standard package. After becoming familiar with it, if you find you need some special capability (within reason), MMS can provide the custom service at a minimal cost.

Typical special Datahandler applications provided by MMS for customers have included special payroll report formats, inventory programs with instantaneous update and report output, replacement of banks of IBM keypunch machines with TRS-80s, interfacing TRS-80s to IBM mainframes, and so on.

Any faults

You can expect that no one program has everything, and The Datahandler is no exception. For example, when working within a Datahandler record, you can back up to a previous field if you have made a mistake. On the other hand, you do not have the capability of scanning records in reverse (although you can quickly scan forward from any earlier record). And, when backing up within a record, each field that you go back through gets deleted and has to be reentered. For example, if you are on field #5, discover a mistake in field #2 and decide to back up, fields 2 to 5 will have to be reentered.

The strong points of The Datahandler, however, outweigh the few deficiencies to be found. The documentation is clearly written and takes two forms. The first, a copy of the Scelbi PIMS Manual, describes a "Personal Information Management System," complete with a program in *Basic* that can be typed directly into a computer. (It is a surprisingly good program despite its slowness and provides an excellent introduction to information management systems.)

The purpose of including the manual is to introduce The Datahandler user to the concept of data management and to provide a beginner-level operator's guide; the PIMS commands described are utilized by The Datahandler. Except for this beginner-friendly feature, The Datahandler is a completely separate program.

The second, a supplement to the PIMS Manual, describes the many unique features of The Datahandler and how to use them. The examples given are particularly useful. In fact, The Datahandler comes with different types of sample files that the new user can load from disk and practice using. The two documents and the sample files provide all the information needed to utilize the many outstanding capabilities of The

Datahandler.

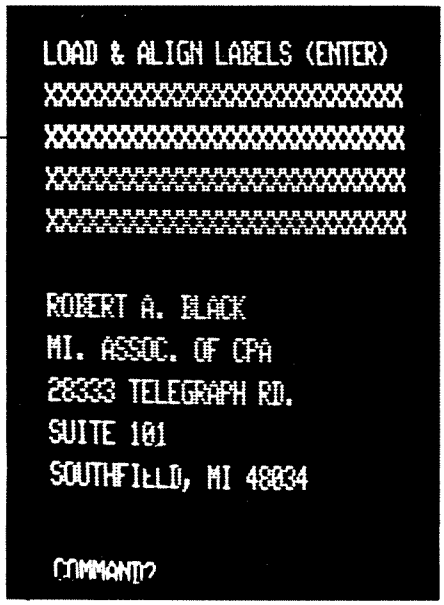
Professional consulting

The benefits of having immediate access to a consulting firm such as Miller Microcomputer Services can not be overemphasized. First, you are working with an established professional and experienced company that has elected to specialize with only a few computers — at MMS the TRS-80 Models I and III and, recently, the new IBM Personal Computer. Because you will have established an enjoyable working relationship with MMS and have come to respect its knowledge and professional skill, in all likelihood a phone call to it will quickly solve a problem.

The MMS programs are solid and proven, hence problems are at a minimum. Among other products from MMS are a General Ledger package and FORTHWRITE, a word processor package designed to use the power of MMSFORTH. FORTHWRITE appears to surpass anything currently available. In addition to being able to use Datahandler files, it is capable of using others such as those created by Scripsit.

An important consideration for a small business is cost, and, when you are shopping around for a firm such as MMS, you must be sure the cost of the various services are up front and understood. MMS spelled everything out, and its pricing and fee schedule were fair and reasonable. MMS's knowledge of the TRS-80 Models I and III and the IBM Personal Computer allow it to advise with a degree of expertise that is seldom found.

The right consulting firm has much to offer the business and professional person looking for reliable software and, perhaps more importantly, reliable after-sale support. □



Mailing label format.

Mr. Johnson, the president of Boston Mailing Company, has worked with micros and their programming for three years. He has had several articles unrelated to computers published in other magazines and publications. You may address any questions to him at 215 A Street, Boston, MA 02210.

Forthwrite

by Wynne Keller

The second generation of word processors is upon us. One newcomer, Forthwrite, deserves to be ranked with the best of them.

★★★★

Forthwrite
Miller Microcomputer Services
61 Lake Shore Road
Natick, MA 01760
$175

Several new word processors have been introduced in the past year. Many reviewers use the term "second generation" to describe these programs. The first generation included the original Electric Pencil and Scripsit. The second generation has arrived with far greater capabilities than the first. Forthwrite is in this new generation, which also includes Lazy Writer, SuperScripsit, and Newscript. All cost over $100 and are intended for the serious writer or business user.

Forthwrite is written in the Forth language and requires the MMSForth disk system to operate ($130 for the TRS-80, $250 for the IBM PC). Forth is a sophisticated, highly flexible language known for its speed. MMSForth has been on the market for several years and is a respected implementation of Forth. Forthwrite is available for the TRS-80 Model I and III and the IBM.

Editing Module

On the assumption that many Forthwrite users are former Scripsit users, some Scripsit commands have been retained, so learning Forthwrite is easy. The same letter abbreviations are used to set margins, turn justify on and off, center the text, and so on. These are entered in a command line within the text, and unlike Scripsit, can be in lowercase.

The editing commands are more logical than Scripsit's, and they are easy to remember. In the TRS-80 version, the clear key is the control key, and all commands function by pressing control and one or two other keys. For example, control I toggles in and out of the insert mode, control D is for delete, and control P for paragraph. Many commands offer additional choices; the delete command, for instance, deletes one letter when first pressed with the option of further deletion of blanks, words, sentences, paragraphs, blocks, or pages, or to the end of text. These additional choices are displayed on the bottom line of the screen, so you don't have to remember all the keys, just the control D.

Those who already own MMSForth will recognize many Forthwrite commands from the Forth editor. Arrow keys provide complete cursor control and can be supplemented with commands for special cursor moves. Nine help screens can be accessed for assistance.

You can reset a number of editing features, such as the cursor's blink rate and speed of movement, the number of type-ahead characters (which allow commands or text to be typed while the computer is busy with other tasks), and the number of lines to be jumped with the shift up and down arrows. Forthwrite provides a screen of dots on which to type; it can be up to 255 characters wide. This helps you visualize layouts. If you don't like the dots, you can substitute another character.

Text blocks in Forthwrite are marked as in Scripsit, but there are many added features. For example, headers and footers can run more than one line. Forthwrite also has many convenient block features; it can get a block rather than just copy it (with Get, you don't need to delete the block from the former position when you are done). You can also save a block to disk, move it to another document, or include it in another document at printout time. The ability to include blocks allows considerable flexibility. You can create a file that calls in other blocks or other documents so material can be reused in multiple documents. Several sample files on the Forthwrite disk illustrate this process.

Column features are very sophisticated. Forthwrite has two types of tabs: visual tabs and printer's ems (measured in tenths of an inch). Visual tabs are displayed on a bar at the bottom of the

screen, as in Scripsit. A printer's ems tab appears at printout time, but its position isn't shown on the screen while you're typing. Either system is easy to use, but the ems are more flexible and give true proportional appearance on the letter-quality Spinwriter-type printers and the new class of under-$700 dot-matrix proportional printers.

Forthwrite's left-margin and paragraph-indent features produce professional-looking documents. By adjusting these settings, single-word left columns can have an adjacent right column of several lines, indented automatically; you don't have to be concerned about where each line in the second column will end. In many word processors, this type of layout requires resetting the left margin frequently. With Forthwrite, you only adjust the setting once, before starting the columns.

The directory is alphabetized for easy, sequential loading of documents with shift, control, and down arrow and for automatic printing of a range of documents. The directory is always available and file names can be capitalized, lowercased, or a mixture. When asking for your file, you do not have to match the uppercase/lowercase display of the directory file name. Type the letters in any form, and they are adjusted to match the directory configuration. If you forget to save the file before leaving the program, a reminder appears on the screen.

Complete find, replace, and wipeout functions are available. These are indifferent to uppercase/lowercase mismatches and have a wildcard capability so you can ignore nonmatching characters or control codes associated with the search word. The replace function can distinguish whether it is replacing a capitalized letter or a lowercase letter, so it can globally replace a single word, capitalizing the first letter if the word begins a sentence. Find, replace, and wipeout can be done on a global basis, with or without a verification before each change.

Hyphenation is often unnecessary with right-justified proportional print, because words are evenly spaced on the line. However, long words may need hyphenation. Forthwrite provides a soft hyphen that is placed while typing with control J. During the printout, if the word falls on the right margin, it is hyphenated. If the word occurs within the line, the hyphen does not appear. A hard hyphen, which is always printed, is also available.

One advantage of the soft hyphen is that rehyphenation isn't needed if you change the document's margins. In Forthwrite, you can't see which words fall on the right margin until a printout is made, because paragraph indents do not occur on the screen, and true proportional print cannot be duplicated with the computer character set. So, you should either hyphenate long words with soft hyphens as you type or obtain a printout and choose those lines that you want hyphenated. Typical proportional print documents require about two hyphens per page.

While editing a document, you might receive the error message "This command illegal when in the white space." Forthwrite calls all the areas of the screen where typing has not occurred white space, since the background characters are white dots. As you type, each letter replaces a dot. However, if you decide to use the down arrow to drop the cursor onto the white dots and add a word to the end of a previous line, the cursor must occupy the space adjacent to the last letter. If the cursor is on a white dot, you can't type until it is moved or the text line is opened with control O. This is disconcerting at first, but it's not difficult once you are accustomed to it.

Unfortunately, you can't determine the line number of the cursor. This is a handicap when trying to decide if a letter, for example, would fit on a single page. Cursor line numbering becomes meaningless if a document uses included blocks, but it would be useful for simple documents. Information on the number of characters in a document is provided, as well as the amount of memory available.

Printouts

Forthwrite has several printout modes. While typing and editing a document, a screenprint is always available. This duplicates the screen and does not format the document.

Quickprint is a formatted printout of the document, and it is slightly faster than a regular printout because menu choices are not required. It retains the document in memory while the printer program overwrites the editor program. After a Quickprint, the document is still available in memory.

A third printing method is a technical copy of the document as it appears on the disk. This method does not format the document. It includes all printing and formatting commands and is especially useful when learning to use Forthwrite.

Another interesting option is to "print" the file on disk rather than paper. In this mode, the file is formatted as it would appear on paper, and it can be scanned for proper appearance. The idea is to save paper, noise, time, and to permit proofing even when no printer is available. I encountered two problems. One is that a document that fits in memory as a document might not fit when formatted. You might have to artificially break up the document to use this option. The other problem is that this feature cannot work if you use proportional type, since these characters can't be accurately displayed on the screen. However, if the document is to be printed in standard pitch, the feature displays margins and page breaks as they will appear on paper.

The fifth and most-used choice is a formatted printout of the document. This has a number of options. First, you should select a printer driver. Special drivers are available for the NEC Spinwriter, Epson MX-80, Daisy Wheel II, NEC 8023, and the C. Itoh Prowriter, in addition to a standard serial and parallel driver that works with any printer but does not provide special features. Other options include multiple-copy printing, the page number to begin printing, and the page number for headers and footers to start printing. You can change the print buffer size, the baud rate (for serial printers), and the number of characters per interrupt, or you can request pauses for inserting text from the keyboard or changing print wheels. Not all these options are available from the printer program; some are part of a customization routine, and many are commands within the document.

One of the nice features of the system is its ability to embed special commands that only affect the printout if the proper printer is available. Otherwise, the program ignores the commands and prints the text normally. Special items such as condensed print, wide print, red-ribbon mode, emphasized mode, subscript, superscript, legal strike out, and underlining are all supported if your printer can do them and you have a printer driver. Because Forthwrite source code is provided you can adjust the existing printer-driver code tables for other printers.

Forthwrite supports true proportional print with right-justification and tabbing on capable printers. A complete test program is included for adjusting your type. The preset width of type characters can be changed. True proportional print means the individual characters are proportional. The j, for example, occupies less space than the w.

The spaces between words are adjusted in fractions of a space, resulting in a pleasing text.

Special Features

Forthwrite can read MMSForth's Datahandler data-base program files, and you can include their fields as text blocks in documents. Format features include right-justification within a field and dollars and cents decimal point alignment, with dollar sign and commas. Names entered into the Datahandler in last-name-first, first-name-last order can be reversed for Forthwrite printouts.

You can dictate a document and then use the Xscribe function to trigger the cassette port to turn the tape recorder on and off while typing. This is not as elegant as foot-activated transcription machines, but it is useful and is a nice "extra."

I have had MMSForth for more than a year, and it is totally incompatible with most other Model I and III programs, since it is not just a different DOS but a different language. Yet, since it is a fine system and is pleasant to use, I am tempted to say it is not out of step with the world, the world is out of step with it.

Gradually, many specialized application programs are becoming available for the MMSForth system, and these programs are made to be compatible with each other wherever the compatibility would be useful, as it is between the Datahandler and Forthwrite. The problem is that this is a slow process, and some needed programs are not available.

The obvious omission here is a spelling checker. Spelling programs are becoming popular, but with Forthwrite, you can't use any spelling checker currently on the market.

Forthwrite provides a transfer module to take any standard TRSDOS (or IBM PC) file and move it to Forthwrite. This is intended for moving files from your old word processor to the new one. A program is under development to effect a transfer in the other direction—onto a DOS; when it becomes available, the compatibility problem will be solved. Using a foreign DOS spelling checker will still cause some problems, but it can be used.

Curiously, Forthwrite provides extraordinary compatibility between the Models I, III, and the IBM Personal Computer. If you own all versions of the program, you can freely exchange files, on the same disk, among all three computers. By using Forthwrite, a modem, and the Forthcom communications program, you can send a perfectly formatted letter at 1,200 baud to any of the three machines. You can also send Forthwrite files to any brand of computer with Forthcom. However, they have to be edited, after they are received, to make them compatible with the format commands used by the receiving computer. If you want to do word processing on your microcomputer at home and transmit the results to a mini or mainframe computer at work, you should enjoy this feature.

The Forthwrite program provides sophisticated printer control, fast operation, powerful include functions, Datahandler compatibility, on-screen help, modular source code that can be altered by the end user, and much more. It can rightfully take its place among the top-quality word processors for the TRS-80 and the IBM PC. ■

MMSFORTH WE'RE READY WHEN YOU ARE!

PRICE LIST
MMSFORTH & ACCESSORY PRODUCTS

(January 1984, available from MMS or selected dealers)

TRS-80 Model 1 V2.0 MMSFORTH Disk System* (32K RAM, 1 single-density drive)	$129.95
TRS-80 Model 3 V2.0 MMSFORTH Disk System* (32K RAM, 1 drive)	$129.95
IBM Pers. Comp. V2.1 MMSFORTH Disk System* (32K RAM, 1 drive, 80-column display)	$249.95
MMSFORTH UTILITIES Diskette Floating Point Math & XREF; also Z80 Assembler for TRS-80, or Turtle Graphics and 8087 support** for IBM PC (32K RAM, 1 drive)	$39.95
MMSFORTH GAMES Diskette BREAKFORTH, CRASHFORTH, CRYPTOQUOTE, FREEWAY (TRS-80), OTHELLO, TICTACFORTH (32K RAM, 1 drive)	$39.95
FORTHCOM Communications Module (48K RAM, 1 drive)	$39.95
THE DATAHANDLER (32K RAM, 1 drive)	$59.95
DATAHANDLER-PLUS**,*** (128K RAM, 2 drives)	$99.95
PIMS Manual (separately) Included free with THE DATAHANDLER.	$11.95
FORTHWRITE Word-processor System (48K RAM, 2 drives)	$175.00
MMSFORTH GENERAL LEDGER System*** (48K RAM, 2 drives)	$350.00
TRADESHOW smart-terminal program*** for Commodities Exchange users (M.3 with 48K RAM, 2 drives)	$995.00

* - General compatability is reported on Dick Smith 80-/PMC-80/Video Genie, LNW-80 (M.1); TRS-80 M.4 (M.3); Columbia Data Systems PC, Compaq, Corona PC (IBM V2.2**).

** - Requires MMSFORTH V2.2***, in beta-test for IBM PC at $15.00 additional. V2.2 supports unusual drives, IBM XT and other hard disks, 8087 fast-math chip, print spooling, virtual disk, and many other new features.

*** - Scheduled for mid-1984 delivery; present MMSFORTH users and clients are invited to contact MMS for beta test-site copies at list price.

SOME OF OUR OTHER FORTH-RELATED LITERATURE

STARTING FORTH (L.Brodie) The best companion book to our MMSFORTH USERS MANUAL.	softcover: $17.95 hardcover: $21.95
MMSFORTH USERS MANUAL (Non-user edition) Intro. to MMSFORTH, but w/o Appendices.	$17.50
THREADED INTERPRETIVE LANGUAGES (R.Loeliger) Advanced; on the creation of a MMSFORTH-like Z80 language.	$20.75
79-STANDARD MANUAL Official definition of the Forth International Standards Team wordset.	$13.95
INTRODUCTION TO FORTH (K.Knecht) BASIC-user's introduction to MMSFORTH V1.9.	$9.95
PROGRAM DESIGN & CONSTRUCTION (D.Higgins) Good introduction to structured programming.	$16.00
TRS-80 ASSEMBLY LANGUAGE (H.Howe,Jr.) Good overview of Assembler and firmware.	$9.95
MICROSOFT BASIC DECODED & OTHER MYSTERIES FOR THE TRS-80 (J.Farvour) Excellent for TRS-80 M.1 firmware; much of it good for M.3, also.	$29.95
THE 8086-8088 PRIMER (S.Morse) Assembly-language view of the IBM PC's 8088 chip, by one of its designers.	$11.95

DEALERS

MMS solicits the participation of reputable TRS-80- and IBM-knowledgeable dealers who are willing to actively support MMSFORTH and its local user community.

LICENSE AND GUARANTEE

MMSFORTH programs require an appropriate MMSFORTH System. Each is sold on a nontransferrable **personal** (single-system, single-user) license basis, with second-part instructions sent upon receipt of the properly-completed License Agreement.

Multiple use: an initial MMSFORTH **Corporate Site License Extension** (CSLE) is available for $1,000 more than the personal license(s), and additional sites or levels of coverage are $500 each (or ten CSLE's for $5,000, then $2,000 per additional ten). The primary level of coverage includes use of the MMSFORTH System, Utilities Disk, DATAHANDLER, FORTHCOM and the GAMES Diskette. FORTHWRITE, GENERAL LEDGER and TRADESHOW each requires a CSLE of its own.

MMS does not issue refunds, warrantee fitness for a given application, or underwrite user losses. MMS does guarantee a machine-readable copy of each program and will attempt to provide reasonable levels of support information.

BULK LICENSING

MMS offers several possible methods of broader distribution of your products built upon MMSFORTH. Most serious applications are delivered with a complete MMSFORTH System, rather as if you had built on CP/M, etc. You would have your end user buy our part from MMS or a MMS dealer, or you may prefer to become a MMSFORTH dealer and offer it yourself, keeping the dealer discount in exchange for your overhead of stocking and support.

If the final application will not require user access to our wordset (no editing, etc.) we can arrange for distribution of a "stripped" version or run-time package. Typically, this means the user can use your words but not ours. Pricing to corporate site licensees starts at $500 for 50 licenses, or for $5,000 you can produce unlimited copies of appropriate programs built on one of our systems -- obviously, very inexpensive in large runs. Wordsets from our application programs are separately available.

MMS ORDERING INFORMATION

Shipping, U.S., Canada and Mexico: (First Class Mail or UPS) is $2.00, plus $3.00 for MMSFORTH System, plus $1.00 for each additional book.
Foreign except Canada and Mexico: add 20% estimated shipping/handling.

Massachusetts orders: add 5% tax.

Prepayment with check payable on U.S. bank, COD, VISA, or MasterCard accepted (include card number and expiration date). No unpaid purchase orders.

Company purchase orders also must be prepaid or UPS COD; must waive any otherwise included "terms and conditions"; must specify whether to be **personally licensed** to a designated user or **corporate site licensed** to a designated building via a designated manager.

Type or print carefully; include phone number. For software: include hardware specs., plus version, serial number and date of latest rewrite (on write-protect tab) of any existing MMSFORTH products. If in doubt, phone MMS for assistance in ordering (MMS will call back collect, if necessary). Prices, specifications and availability are subject to change.

MILLER MICROCOMPUTER SERVICES
61 Lake Shore Rd., Natick MA 01760, USA (617/653-6136)

CREDITS

WHO USES MMSFORTH?

There is no one typical user group of MMSFORTH. But, with changed names, here are a few descriptions of real users:

Joseph S. bought an early TRS-80 Model 1 microcomputer, learned BASIC and bought one operating system after another in the search for a better software environment. His sales representative job benefited from his use of a few application programs (database and wordprocessing), but he became challenged by the programming itself and quickly found BASIC to be limited in its abilities and ease of use. He tried FORTRAN and Pascal (with some success, but not with enthusiasm) and had a bruising round with Assembler. Joe says, "Some of MMSFORTH is quite advanced, but the part I wanted proved surprisingly easy to learn; it's the environment I was waiting for." He bought THE DATA-HANDLER primarily for an inexpensive practical example of Forth in action, but was surprised to see it displace his more expensive - and far slower - database system for business use. Now he can modify its source code for his own custom jobs, and he feeds its data into FORTHWRITE for professional, proportionally-printed personalized mailings. Joe likes our upgrade policy, too: he paid $80 for his original, personally-licensed MMSFORTH V1.8 Disk System, plus $10.00 rewrite charges for updates (plus the difference in price when the V2.0 list price became $130). He paid $60 for THE DATAHANDLER and $175 for FORTHWRITE, and says, "The combination beats a popular set which doesn't provide equal performance or as good a match, but added up to $400 without an operating system or language."

Prof. Jonathan C. has been adapting TRS-80 microcomputers, and later IBM PC's, for laboratory and shipboard research use. In past years he invested many thousands of dollars in commercial languages, operating systems, utilities and internally-developed software to meet his varied and demanding needs. Now he says he no longer buys any other software at all, as MMSFORTH has given him an environment that grows with his needs and transfers existing program modules to new jobs. (He did buy another MMSFORTH System for his PC, and found the transfer very easy.) The compactness and speed of MMSFORTH permit big jobs in small computers. "**Forth,**" says the professor, "**is the language that microcomputers were invented to run!**" And for the really big jobs? He finds that his dedicated IBM PC, running MMSFORTH with 8087 (fast-math) co-processor support, completes many projects faster than he can with his university's CRAY-1 - after accounting for turnaround time on the overscheduled mainframe - and that the other jobs are easily transmitted across campus using our FORTHCOM communications package; its source code permits him to make many quick and effective modifications for his changing needs.

Roger B. is in charge of the software development group at a major manufacturer of dedicated computer hardware systems. Traditional assembler programming had to give way to something offering more efficient development and maintenance, so he hired MMS to assist in selection of a suitable new language for their large projects. We stayed to train the staff in Forth, and to create certain critical modules of their time-dependent programming. At relatively minor cost, MMS helped Roger's company stay at the cutting edge of its multi-million dollar business.

John T. is an accountant who has expanded his out-of-the-home business by adapting the General Ledger in MMSFORTH for his own needs and those of his clients. He likes its reports flexibility for breaking out sub-headings and mixing various journals and ledgers. But the best feature, in his opinion, is the very high level of keyboard automation which allows his practiced fingers to enter twice as much data per hour as with most microcomputer accounting packages. Now John is recommending that his clients get their own computers and General Ledger programs to cut down his processing time. (He won't lose money because he can handle more accounts.) If this new aspect of his business keeps growing, he may become a MMSFORTH dealer and provide the software directly to his customers. His individual clients, like himself, buy the software under personal license for $480 ($600 on IBM PC); corporate site licensing costs an additional $1,500.

Susan W. does wordprocessing on an IBM PC at work, and had tried several other programs before using FORTHWRITE. She finds it very fast (she had to slow her typing and thinking for some of the others), able to include boilerplate sections, and with her company's preferred formats for print-out in executive-look, full proportional type - even for multi-column tables. She values its large capacities: it stores 395K per 40-track, double-sided drive compared to the standard 360K, and it can continuously print a range of documents from its alphabetized directory, spanning multiple diskettes when necessary for very large reports. Sue particularly likes FORTHWRITE's sensible letter-key mnemonics ("They're so sensible you can guess them in advance!"). While she was learning, she also appreciated the bottom-line prompts, built-in help screens, good manual and sample documents. Sue's boss often works at home using his own FORTHWRITE on his TRS-80 Model 3; next morning, he pops his data disk into her IBM PC for her direct use. The cost? Her evaluation copy (on non-transferrable personal license) of FORTHWRITE and the MMSFORTH Disk System cost $425, but the company subsequently decided to buy Corporate Site License Extensions for her whole office building for an additional $1,500. (15 computers and about 25 persons presently use the software at an average cost of under $200 per computer, and many more may be added at no additional cost in the future.) Her boss bought his home software under a personal license for $305. (Like his computer, it's tax-deductible.) When their company set up the full-office operation, it also hired MMS to run a one-day FORTHWRITE workshop. They could have managed without it but found that the training was cost-effective, it assured user acceptance, and it rapidly increased the skills level. For the same reasons, the company maintains a client contract with MMS, with Sue as the designated contact between all the company's users and MMS. First-year consulting costs, including the workshop, were under $600. The company is considering adding FORTHWRITE installations in several other buildings in other states. Licensing each new building will cost $500 less than the first building. Some of the biggest bargains come later: when they add telephone modems, FORTHCOM communication software to transmit "electronic mail" between locations will only add $40 per building. This can include FORTHWRITE documents and DATAHANDLER-type files, so they intend to add DATAHANDLER-PLUS for another $100 per building.

Sam G. runs a radio and television repair shop. His attempts to use standard TRS-80 software resulted in a year of waiting. He hired MMS to adapt THE DATAHANDLER to his task and now has a near-immediate ability to track all jobs and to print repair tickets, customer receipts and summary reports. The program works exactly the way he wants and has required no maintenance in several years of operation. It took two weeks to deliver, and cost a total of $500. (Sam, as the only user, operates it on a personal license.) Sam says, "They were right, the computer did make me money, but it took MMS to prove it."

Robin T. develops software for general marketing. His very popular communications package was originally written in Assembler. Then he tried MMSFORTH out of curiosity and said, "Assembler? Never again!" Robin's next version has been written in MMSFORTH, modified for running under DOS. He likes the idea of developing compatible code for the TRS-80 and IBM PC - two quite different computers - in the same Forth, and he prefers MMSFORTH to other Forths because of our better utilities, performance, and ability to work with him on special needs. Robin will be distributing his next-generation communications package with a specially constructed **MMSFORTH run-time module** (a stripped portion of our MMSFORTH System) imbedded. He will pay MMS $500 for rights to sell each 50 copies (i.e., $10 per run-time module), or $5,000 for unlimited copies used with one or more of his own products. Since he is working alone, he does not have to pay the extra $1,000 for his MMSFORTH corporate site licensing until the product is ready to distribute.