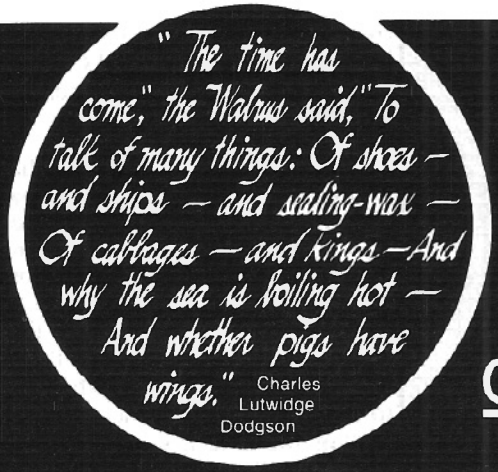
$5.95 U.S.

two/sixteen magazine™

Volume 1, Number 2 July-August, 1982

The journal for business, professional, and scientific members of the TRS-80 community



CONTENTS

two/sixteen magazine

Editor/Publisher — Richard H. Young
Associate Editor — Barbara S. Albert
Art Director — Timothy P. Schleif
Contributing Editors — Arnold Fischthal, John Thornton
printed by — Science Press, Ephrata, Pa.

two/sixteen magazine is published bi-monthly by
two/sixteen
131 East Orange Street, Lancaster, PA 17602

Telephones

Editorial staff: **(717) 397-3365**
Classified advertising: **(717) 397-3789**
Display advertising: **(717) 397-3364**
Subscriptions: **(717) 397-3364**

Annual subscriptions: $30 in U.S.
VISA and Master Card accepted.

Foreign subscriptions:
Canada: $36 U.S.
Other foreign: $47 U.S.
Air mail rates: $67 U.S.
Drawn on U.S. banks.

Send all subscription correspondence, including change of address (with label) to
two/sixteen
Box 1216, Lancaster, PA 17603.

two/sixteen subscriptions may be obtained at over 1000 computer stores in the U. S. and at any of Vijay Stores' three convenient locations in India:

62, Kalyan Bhuvan, Relief Road
Ahmedabad—380 001

231, Dr. D. N. Road
Bombay—400 001

Station Road
Anand—388 001 (Gujarat State)

Editorial

We've had a lot of pleasant surprises since beginning this effort to assist Model II and 16 users.

First, we now have paid subscribers in more than 25 countries, including such far-flung locations as Papua New Guinea, Singapore, Kuwait, and Sweden (Our New Guinea subscriber reports that his Model II works very well except during the rainy season, when condensation seems to cause malfunction).

Acceptance of the magazine has been excellent. We have received many letters and phone calls praising the magazine. Many subscribers have also offered constructive suggestions for improvements, some of which we have been able to incorporate in this issue (one excellent suggestion, that we include a glossary of terms which are likely to be unfamiliar to some subscribers and to flag these terms in the text so that readers will know that they are in the glossary, turned out to require too much time and effort at the critical point in production, so it has been deferred for an issue or two). Only one of the first thousand subscribers has cancelled with disapproval. A gentleman on Guam wrote "I am not impressed with either the appearance or the content of your publication at this time. Please CANCEL . . ." We wish that he had given us the benefit of knowing what he wanted to see. A couple of others cancelled because they had given up on Radio Shack (We're really sorry to hear this; if they change to other computers for business use, they'll almost certainly find their situations more frustrating).

I'm very impressed with the high level of satisfaction that II and 16 owners have in their computers. You'll see, in these pages, quite a few negative (but, we hope, constructive) readers' criticisms of various aspects of Radio Shack (software features, manuals, lack of knowledge on the part of their salesmen, service policies, etc.), but almost universal acclaim for the computer itself. Many of us believe that the II and 16 are the best business computers available today, although many of us use CP/M,

OASIS, TURBODOS, or other non-Tandy operating systems. Tandy acts as if it resents these other systems, yet the flexibility that the II and 16 have in being able to run under more operating systems than any other computer is one of the things that make the II and 16 the best (e.g., it is now possible to buy a board that will allow your II or 16 to run any of the rapidly expanding repertoire of programs that run on 8086/8088 CPUs; e.g., the IBM PC). We hear a lot of speculation to the effect that Tandy will be improving its policies with respect to at least some of the other operating systems. We hope it is true, as this should result in much greater sales of IIs and 16s, which, in turn, will lead to more resources becoming available to support all of us with new software, from both Tandy and outside software developers.

In this issue, we have published most of the letters received in the last two months. We did this so that readers could get an idea of what their counterparts are experiencing. We'd like reader comment about this policy, because we expect to receive substantially more letters than we will be able to publish. Are these letters (and our responses) in tune with your needs? Should we more thoroughly edit the letters so as to reduce extraneous comment? Should we print fewer letters or more? Should our comments be different in some ways?

Several people have written to ask us for free copies of the magazine. One subscriber suggested that the first issue should be free to everyone, since it was "introductory." Two or three readers have expressed the view that software, in general, should be free. We are somewhat mystified by the reasoning that must behind lie these requests. Publishers and software developers are in business, just like most of our readers must be, and must meet payrolls, rent, printing bills, etc., just like other businesses. I would be delighted to receive a grant from an individual, a foundation, or even the government, so that we could provide free copies to everyone who is interested in IIs and 16s. Until such manna is forthcoming, however, it will be our policy not to provide free copies (except to bona fide potential advertisers) at the expense of the 8% who support us through their subscriptions.

As to software, I believe that software developers should be fairly compensated and that the fruits of their intense labors should not be stolen from them by software pirates. Anyone who believes that good software should be free has obviously not developed any himself. Some who have written software for their own use mistakenly equate their effort with that required to produce a marketable product. From my experience, I'd guess that the effort to produce a generally marketable product that is based on a successful one-user program is at least ten or twenty times the effort required to produce the original program. In addition, the product must be marketed so that users will know it exists. Advertising is very expensive. For every Word-Star or VisiCalc, there are hundreds, even thousands, of well-written, useful, software products that never reach the break-even point in the marketplace. Because of the potential volume in the micro world, programs are very inexpensive compared to similar programs for the big computers. Micro programs seldom cost more than 1% or 2% of essentially equivalent programs for large mainframe computers. So please give the developers a break!

If there is anyone who doesn't agree with the moral principle involved, then look at it from a purely selfish view: if developers can't get a fair return for their software, there just won't be any new software. It's just like any other business. As to pirates, if we ever find that one of them has advertised in our magazine, we will spare no effort to aid in their prosecution. I believe that commercial software pirates belong in the same anti-society category as muggers and heroin-pushers. On the other hand, we hate to buy software distribution disks only to find that they have been protected by one scheme or another and cannot be backed up. Every time we use such a program we are required to subject the original distribution disk to the possibility of damage by a disk drive (see the article, "Barbara's Adventures . . ."). This seems silly and counterproductive; pirates are often experts at getting around these protections and will do so in short order; the only ones inconvenienced by the protection are honest purchasers.

A number of subscribers have thoughtfully inquired as to how we're doing (meaning, will the magazine survive?). My answer is "yes, I'm reasonably sure that it will." However, to survive in the "long run" (in fact, to break even), I estimate that we need 4000 to 5000 paid subscribers. Right now, we have about 1000, and we're growing at 200–300 each month. In addition, we distribute over 1200 copies to dealers and potential advertisers, in the continuing effort to obtain subscriptions through dealers and advertising revenues from product vendors. Our main difficulty is reaching those tens of thousands of Model II and 16 owners who have no way of knowing about our existence. We tried a mailing to about 300 computer clubs and got almost no response at all. We advertise in *80 Micro* every month, and receive a reasonable number of subscriptions as a result. We have also used a mailing list purchased from a software vendor, and got a good return from that. However, Radio Shack has a policy of not renting their mailing lists (or even letting anyone know how many of their Model IIs or 16s have been sold). In fact, we are unable to find out who has these machines in our local area, and have not one single subscriber among the presumably large number of Model II owners in Lancaster (although we have subscribers in Hong Kong and Maracaibo). If subscribers are aware of the existence of Model II (or 16) mailing lists for rent, please let us know about them. We are interested in any list of 100 owners or more, and will pay $60 per thousand names on peelable labels, $70 per thousand on "Cheshire" labels, or $90 per thousand on diskette (ASCII, in TRSDOS or CP/M format, with either comma-delimited or fixed-length fields, and either carriage-return-delimited or fixed-length records). Other suggestions for increasing circulation will be most welcome.

Readers are welcome to discuss their problems with me in person or by telephone. If you vacation in Pennsylvania Dutch country, please stop by. Our offices are just two blocks from the center of downtown Lancaster, and we're officially open from 9 am to 5 pm, Monday through Friday (although you are likely to reach me here almost any time by telephone). RHY

Lift up one hand to heaven and thank your stars if they have given you the proper sense to enable you to appreciate the inconceivably droll situations in which we catch our fellow creatures.
—Sir William Osler
LIFE OF SIR WILLIAM OSLER

The most powerful weapon of ignorance—the diffusion of printed matter.
—Count Lyof Nikolayevitch Tolstoi
WAR AND PEACE

Letters to the Editor

Our policy is to publish all letters on topics which may be of general interest to our readers. We have included most of the recent letters received so that our readers may see the cross-section of other readers' concerns. Our comments, in italics, follow many of the letters.

We are particularly grateful to the many subscribers who took the time to write to us about their experiences and problems. Several patterns have emerged that will help us provide Model II and 16 users with useful reports.

Several of the following letters ask for assistance beyond the knowledge of the two/sixteen *staff. Readers with solutions to any of these problems are urged to respond to the writer directly with a copy to us.*

From Roger Cruon, Consultant, L'Amiral A.-Av. Tremolieres 83160 La Valette-du-Var (France):

I was glad to receive your first issue and I think the idea behind it is excellent. I hope you will receive enough material to expand it soon. Here are some comments on one of the papers published.

All the points raised in the paper "Lamentations on the RS BASIC Interpreter" are very well taken. I disagree, however, with the conclusion that the only alternative is to use other languages. There are some means to remedy, to some extent, the shortcomings of BASIC, using a combination of programming tricks and utility programs.

The main limitation, and I would say the only serious one, is the lack of "local" variables (point 4). I have no remedy to this situation, but it may help to reserve to standard subroutines the variable names formed with one letter and one digit, while using two letters names for normal coding. The use of an online cross-reference utility such as the one contained in SNAPP-II extended BASIC is also vital. This utility prints all references to a given variable, if any, in a fraction of a second, even in a 20k program.

Generally speaking, I highly recommend the line of extensions to the RS BASIC interpreter provided by SNAPPWARE; a set of 16 patches to TRSDOS, including the 4 you listed in your magazine, is also provided free of charge to customers, although they say this material is copyrighted; it thus seems that SNAPP Inc. deserves acknowledgement for them.

Here are a few of the capabilities provided by the SNAPP extensions; only those having a direct bearing to the points raised in your paper are mentioned.

The cross-reference utility already mentioned also gives the references to lines (point 5) and strings, and may even make global changes, but only replacing a BASIC keyword with another (point 7). The disk I/0 management facility (SNAPP-IV) allows for a formatted reading of direct access files (point 8). A renumbering facility allows relocation of program code (point 7), while a compressing facility with various options may be used to obtain a working version of the program where comments are suppressed (point 6) and program length is further reduced by various tricks.

There is also an utility from RACET Computes, called BLINK, which provides a chaining capability and preserves all current variables. I did not find it as useful, however, and I usually prefer to divide the code in separate programs, each with its own initialization of variables.

As far as the execution speed is concerned (point 1), the main limitation for business software is generally the access time of floppy disks, which is language independent. For scientific computations, the use of the Microsoft Compiler, a real compiler, may be the solution.

I hope these comments may be useful to your readers.

I would rank non-portability as a more important shortcoming of BASIC. For example, Model II FORTRAN or COBOL will eventually run on the 16 with litle or no modification (and a simple recompile). I think it unlikely that Model II BASIC will run on the 16 without prohibitively extensive modification.

We discovered, shortly after going to press, that the patches we had published originated with Bob Snapp of Snapp, Inc. They are part of a series of sixteen "DOSFIXes" which Bob has graciously allowed to be reproduced in any form, provided that acknowledgement of the source is included. We are therefore reproducing them elsewhere in this issue under the title "DOSFIXes: Courtesy of Snapp, Inc."

FROM: Acoustilog, Inc., 19 Mercer St., N.Y., N.Y. 10013:

Your composite reader is me, to *A TEE!*

I'd like to see articles on Assembly Language for the Model II, what are its (Assembly's) capabilities and limitations, how *exactly* to key it in. Also, when discussing TRSDOS, please mention which version (I have 1.2). GOOD LUCK!!

P.S. Those 5 patches don't work for TRSDOS 1.2, I'm disappointed to report. All 5 were aborted. Anything that can be done??

You may assume that all TRSDOS references are to the latest release (currently 2.0a for Model II and 2.0b for Model 16) unless otherwise indicated.

We recommend, as do many other writers, that TRSDOS 1.2 users convert to 2.0. This involves some time and expense, but an eventual upgrade is inevitable for all 1.2 users and the sooner it's done the easier it will be.

FROM: Alan C. Ryan, 2126 Haygate Cr., Mississauga, Ont. L5K 1L5:

I tried the Patches shown on page 8. I found Patch 3 should read A=0CFB, not OCFB, etc. Patch 2 was completed OK, but when I tried to copy a file, I got Error 25. I tried using KTA2 as password but no luck. Any suggestions?

I think your first issue is great and look forward to further issues.

Our typesetter made the unfortunate substitution of letter O for numeral 0 in

patch 3. Patch 2 works for some 2.0 users but not for others. We don't know why.

FROM: Gary Boatright, Box 306, Vinita, Oklahoma 74301:

First let me welcome you to the computer magazine market. You are publishing just what I want to read and need to learn. Thanks for writing something for Model II users, like me, specifically.

I have several questions you can help me with: (1) Would converting to COBOL be worth the time and money? Because of my unique business, there are no Radio Shack programs that I want to use. I have written my own and would have to re-write them if I changed to another compiler. (2) Should I try to sell my Model II and get a 16? I need the 2.5 Meg. disk storage and the increase in speed would be welcome also. (3) Is 1985 going to make all of this questioning obsolete? If there is something better than the 16 on the horizon I can well afford to spend the money later, rather than now. Thanks!

(1) I confess to a long-term bias against COBOL. My preference is FORTRAN for any kind of business application. This arises from some 10 years of FORTRAN use on interactive large-scale computers (primarily DEC-10s). The microcomputer environment is somewhat different in that no really good extended FORTRAN has yet surfaced. I suspect that one may appear when the 16-bit operating systems are more mature. Depending on the nature of your unique business, COBOL may do the job for you. Several readers have expressed satisfaction with RS COBOL. The three main objections to COBOL are its extreme verbosity, its awkwardness in an interactive environment, and the difficulty (even, sometimes, impossibility) of handling complex financial calculations. In my experience with large systems, it took several times as long to develop and to maintain COBOl systems as it did FORTRAN systems.

(2) Each Model II owner must work out his own study of the economics of upgrade vs. replacement. A key variable is how much you can get for your Model II. My preference, assuming you cannot afford to keep the II and also buy a 16, is replacement. Whatever the future of storage media, it's likely that floppies will be used for transporting programs for a long time to come. And if you use floppies at all, two drives are essential. The two-drive 16 is the best buy in this year's micro market, in my book, even if you use it only in Model II mode. (However, note that in Model II mode you can get only 500,000 bytes per drive.)

(3) 1985 will make all of today's questions obsolete. But business decisions about computer purchases ought to be based on whether or not the equipment will pay for itself in, say, three years (two years for hard disks). Most good applications will do far better than this, and your purchase decision should not be influenced by questions of obsolescence.

FROM: Paul H. Johnson, 6220 Pridgen St., Panama City, FL 32401:

MISCELLANEOUS OPINIONS: (1) Please publish more practical, useful articles. Include programs we can use in business (including sort routines, mailing lists, etc.). (2) Please don't take up space with wordy, "editorial-type" articles, such as "Comparative Productivity In Computing" (yawn) that appeared on pages 22, 23, & 24 of the first issue. (3) Please do have a "Letters to the Editor" column. (4) Please do have "How-To" articles, such as Beginners Assembly Language, how to use CP/M, etc. (5) Please do have software and hardware (and book) reviews. (6) I would like to see a comparison of the MX-100 Printer, the OKIDATA 84 Printer, and the Radio Shack Line Printer V, when used with the Model II with common software. THANK YOU FOR PUBLISHING *two/sixteen magazine*! WE NEED IT!!

Hey, Paul, some of our readers really liked that article! (Actually, I was a bit surprised myself that anyone did. It was a last minute "filler," done to take up some empty space the printer unexpectedly stuck us with.) However, let me assure you that our objective is to publish primarily practical, useful articles. By the way, while we can usually obtain review copies of software and books at nominal cost or gratis, there isn't any practical way (from an economic viewpoint) that we can obtain major hardware (other than what we purchase for our own use) for review by our staff. Therefore, we are especially interested in getting comparative hardware reviews from readers who have access to more than one variety of printer, disk drive, or other peripheral.

FROM: Robert Streever, P.O. Box 32246, Louisville, KY 40232:

How about an article on how to implement the supervisor calls described in the TRSDOS Manual? I have no idea how to make these things work, and a salesman at the computer center I asked knew nothing at all about them. Also . . . Can you review the Microsoft BASIC Compiler? THANKS—GOOD LUCK!!

Several letters and phone calls have requested help with the supervisor calls problem. I was just as irritated as anyone when I found, after studying the 80 or so pages of the Owner's Manual devoted to these calls, that no way is given to use them. The Shack was no help at all. In fact, when I observed that an Assembler was probably necessary to make full use of these calls, the RS Computer Center was unable to tell me anything at all about the three different Model II assembly packages they sell (they weren't even in stock and would have to be purchased "in the blind" by special order), much less what the significant differences between them are or how one might become proficient in their use. (Tandy please take note: this needs corporate attention.) I suspect that there are several readers out there who are expert in the use of supervisor calls, and I earnestly solicit an article with specific directions (including code).

We've had several requests to review the Microsoft BASIC Compiler. I've also talked with several of its users, all of whom praised it highly, particularly in comparison to other BASICs, both interactive and compiler. We will publish a review later this year.

I grow old learning something new every day.
—Solon
VALERIUS MAXIMUS, Book VIII

FROM: Dr. Gary Turkel, 137 Lafayette Avenue, Suffern, NY 10901:

I have high hopes for your magazine; good luck. My DREAM is to see a commented Disassembly of TRSDOS!! Short of that I'd like to see articles on modifying parts of TRSDOS—the patches were a good start. I hadn't seen two of them. I would like to find out the address containing the code of the parallel printer port. This and some hardware additions will enable me to keep a number of printers "on-line" at the same time. I appreciate good hardware and software reviews. The one on the microbuffer was good. It confirmed my impression as to its limited compatability (Epson only) and time saving.

Wayne Green is a bore. His anti-Tandy attitude is obnoxious and taints the credibility of the entire magazine (*80 Micro*). If his sparring with Tandy doesn't stop in print, I will probably cancel my subscription. It's only of marginal value to me anyhow.

RS's *Microcomputer News* is a little more useful and, of course, is essential for "official" patches.

Computronics MOD II Newsletter is little more than a sales blurb, but it's not without some value.

I would like to see *two/sixteen* switch to lighter weight paper and 3 holes to accommodate a ring binder.

I have a small number of Z80 routines that I call from my COBOL programs. They vary from 4-100 lines and could be incorporated into an article demonstrating the use of TRSDOS Supervisor Calls. If it sounds interesting to you, let me know. P. S. CP/M doesn't interest me.

Assuming that someone (other than Tandy) had the talent to produce a commented disassembly of TRSDOS, it could probably not be published without violating Tandy's copyright. I think you'll find that all producers of operating systems (whether hardware manufacturers or not) are very sensitive about revealing the details of their code. They have a large investment in development, and want to maintain a competitive advantage. This is a misguided thought, as the experts (the competitors they are keeping the secrets from) are able to analyze the operating system without much trouble, while loyal users are unable to benefit from, for example, a commented disassembly.

PLEASE do an article on supervisor calls!

FROM: (Name and address withheld):

This is my first letter to any of my magazines in twenty years of working. I feel you are trying to help what appears to be a relatively small group of beleaguered professionals required to perform miracles with no trainng and little budget. You have all the support I can muster.

I am presently (between normal work chores) trying to fill the gap between a $20,000/month time share useage and the very little we are allowed on our in house HP64. Your dialog on the professional DP organization hit the heart of our problem. Fifteen years ago we began begging for DP help in the Engineering and Quality Control areas of our plants. We gave up and went the time share route. When we passed $20,000/month about four years ago, management pounced on us. "We need that data; but DP can do it for much less"—NOTHING! We expensed some Model I's for trial and then snuck in five Model II's to get going. The drop in time share costs easily covered the expense before we were found out; but politics stopped the complete demise of our time share. We have now "legally" obtained three more Model II's and the time share costs are close to $20,000/month again. DP is again trying to save our souls by a comprehensive Engineering QC real-time/batch system set up to be "user friendly." The present projection is sometime in August of 1984—whoopie. In the interim we are using the Model II's at below capacity to hold down the politics.

The programming I do as part of my job is limited to the R. S. Interactive BASIC to allow the people using the programs to make minor changes without my involvement. Also it allows me to link many programs together for semi-automatic running of various reports.

I have hit a capacity problem on sorting that bothers me though. I use a modified segmented sort with the "swap" command to sort within the Basic programming. I am limited to about 1900 lines at a time and need about 20 minutes to do this. That becomes a pain on groups of 5000 or more and requires splitting and appending. I would like to see something on assembly language subroutines incorporated into BASIC programs (especially sort routines). I need all the help I can get. I bought a Model II and LP VI for use at home; but I can't seem to force myself to get with it in assembly language programming.

Good luck with the magazine. Holler if I can help.

You are fighting the politics of inertia which has become a crippling force in American business. It has become a sadly common event for senior management to condone empire building by "data processors" whose primary goal is maintenance of the status quo. Modern MBA-type education results in management with no practical experience in the actual business of the company, coupled with a cover-your-backside and don't-rock-the-boat attitude. Fortunately, sparked by ever increasing losses of business to the more efficient Japanese, there is a new and growing movement to do away with this nonsense.

Sorting: I recommend SuperSort by MicroPro. CP/M based, it will sort data in almost any commonly-used format (including, for example, fixed, variable, C/R-delimited, Twos-complement, Integer, Floating-point, Packed-BCD). It is by far the most flexible sort utility I've seen, even when compared to sort packages on large mainframe systems. SuperSort can be called as a subroutine from MicroSoft's FORTRAN, Relocateable Assembler, or COBOL. It typically retails for about $200, and is well worth it.

Assembler: Most of us who have an interest in learning assembly language have made many "false starts" in the past. Writing about assembly language often degenerates into a demonstration of erudition by the author. Sometimes it simply shows the author's inability to perceive the audience's level of knowledge. However, we have found a course in assembly language which is designed to bring the neophyte along at his own pace. It is the Heathkit course "Assembly Language Programming," Model EC-1108, which costs $50. Even though it is primarily aimed at the Heath computers with 8080/8085 chips, it will probably be clear to any Z80 user.

FROM: John Doerschuck, Indian Waters Drive, New Canaan, CT 06840:

Congratulations on the first issue of *two/sixteen*.

Most of us fit your "composite reader" spec pretty well in that we've just never gotten the information we could really use for the Mod II. That thick Owner's Manual, excellently prepared and updated at great cost, is mostly useless; I suspect it was prepared for systems programmers working for outside software houses. It took a long time for a book like *Basic Faster & Better* to be issued, and as one might have expected, not by the Shack!

I have purchased three database pro-

grams: The DataBank, The Creator, and MTC's AID3. I'd be happy to review all three, though the first two can be dismissed in a few sentences. How happy (and more prosperous) I would have been to have seen reviews before buying!

Now, possibly like other Mod II owners, a hard disk looks like a good way to speed up file processing. (Most of our files are in the range of several hundred records of perhaps 120 bytes each.) So a review of hard-disks is in order. Many of us have service contracts, and what the RS service man says about voiding the maintenance contract doesn't agree with what outside vendors tell you about service problems. I just got a demo of the RS hard disk, and was not terribly impressed.

How about the outfit that will retrofit your Mod II to use RS memory cards and have 128k of memory? Is this feasible?

As I see it, the ONLY reason I would want to upgrade to Mod 16 is to obtain direct access of more than 64k, and possible attendant speed increases, which are probably of minor benefit to single-user systems. So *two/sixteen* may end up talking to two different user groups: single users and multi-users—two very different populations.

Intelligent reviews of CP/M vs. TRSDOS would be appreciated: should I switch or shouldn't I?

We'd like to get your review of the three data base systems.

As to hard disks, see our article in the current issue.

We've seen the specs on several of the "memory upgrades" but have not seen one in action or heard from anyone who has one. Be warned, however, that the TRSDOS addressing scheme (and all the other 8-bit machine addressing schemes we know of) can address only 64k of memory. Additional memory seems always to be used in one of two ways: (1) as bank switched memory, allowing more than one job to share the CPU while each has its own bank of memory or (2) as a pseudo-disk with very high speed access.

The Model 16 upgrade will improve performance of existing *programs probably only if they are in RS COBOL (they can be recompiled, they say, in Model 16 RS COBOL without source language changes) and you do a lot of sorting. Of course, if you are disk-I/O-bound now, you will almost certainly not gain much, if any, from conversion.*

Many use both CP/M and TRSDOS, and the reason for CP/M is the overwhelming array of fine programs available in CP/M, as compared to the handful in TRSDOS. We'll have detailed comparisons in a later issue, but program availability is what sells most of us on CP/M.

FROM: Charles Z. Tzinberg, ShopLand, Inc., 6930 W. Main Street, Belleville, IL 62223:

I will eagerly await the delivery of my first issue of *two/sixteen magazine*. I started with a TRS-80 Model I when they were first introduced, and have continuously upgraded to my present Model II with Cameo hard drive, Daisy Wheel II and Line Printer V.

I subscribe to many computer magazines, but of course the amount of usable material I find has decreased through the years until now I am content with a small article now and then which is interesting and usable. Even the *H & E Computronics Model II Newsletter* is extremely disappointing.

FROM: Peter G. Jensvold, 234 Whitney Street, San Francisco, CA 94131:

OK, Richard—I'm hooked for two years. I have a Mod II 64k with 2 drives and will probably go to the 16 with hard disk soon. We need *two/sixteen*.

I'm sick and tired of reading ads only to find a program/system is only available on a 5¼" disk! GOOD LUCK!!!

FROM: J. O'Loughlin, Micr-Innovations, P.O. Box 53, Huron, OH 44839:

I have a few random comments about things I saw in your first issue. Naturally I haven't made any attempt to organize them but I thought your invitation to comment was too much to resist.

1.) P. 24—DBMS comment—I too share your frustration about the lack of a "good" host language interface, BUT wait a minute. There is at least one vendor that has the ability to both generate and accept ASCII files via their DBMS—Who?—Condor. It certainly isn't the same answer that the DBMS systems for the biggies have but it isn't nearly as restrictive as dBASE II, for example.

2.) P. 17—The TRSDOS CP/M conversion—count me interested.

3.) I have some of the general utilities from RACET, SNAPP, the MICROPROGRAMMERS. I could certainly comment about these from the user's point of view—Which would you like to see articles on?

4.) Wish list—After many frustrating hours, I finally managed to figure out how to make things talk to Serial Port B. I thought if I could figure that little item out I would like to figure out how to talk to the disk. What is so special about my little Port B program—I DID NOT use any SVC calls. Mine is coded in assembler. I want to do the same thing for the floppy disk and I think (?) I have a pretty decent starting point from what I learned on the Port B exercise. At any rate, I sure would like to see something that goes at the problem from BOTH the SVC and direct coding point of view.

5.) I share your view of the Basic Interpreter, BUT, the outfit called SNAPPWARE has a set of enhancements for the interpreter that vastly expand the capabilities of the animal to the point where it certainly is much more useable.

6.) Want to Compile your BASIC code—forget the Shack product and get BASCOM from Microsoft. It will certainly answer most if not all of your concerns.

Other things: I have written several articles, short to be sure, but they made it to print, and I believe I may be able to contribute to your effort. I have the usual collection of credentials but perhaps the most important are the two years of heavy experience with the Model II and a total of 6 years ongoing experience with timesharing systems such as the PDP11/70, DEC-20, and HP Series 3000.

Thanks for the opportunity to tie up your time for several minutes. I appreciate your efforts on the new magazine and am looking forward to future issues.

(1) How about writing an article comparing Condor and dBase II?

(2) It turns out that there is a two-way TRSDOS-CP/M conversion package available. It is one of the ReformaTTer series from MicroTech Exports, 467 Hamilton Avenue, Palo Alto, CA 94301, telephone (415) 324-9114. It may seem expensive since, at $249, it costs more than a CP/M operating system. However, it includes many useful functions in addition to two-way transfers. We'll be reviewing it soon.

(3) We'd like to see articles on all of these. Even though we are publishing an article on Snappware in this issue, another viewpoint is always welcome.

I was gratified to be able to answer promptly, and I did. I said I didn't know.
—Mark Twain
LIFE ON THE MISSISSIPPI

A Review of Snapp-Ware's SNAPP-II

By Dr. Arnold H. Fischthal
184-08 Tudor Road
Jamaica Estates, New York, 11432

Snapp-Ware of Cincinnati, Ohio produces a series of software products that makes the life of a program developer almost bearable and, at times, even enjoyable. One of their products is a functional enhancement for the BASIC Interpreter called SNAPP-II Extended BASIC, which is an extremely useful aid in program development and debugging. I have used this product for almost two years, and it has become indispensable.

SNAPP-II is quick and easy to install on any system disk. In fact, total installation time, including reading the manual to make sure you are doing everything correctly, is under two minutes. All you really have to do is make several patches to the BASIC interpreter so that the new commands introduced by this product will be recognized. The software takes up no user space in RAM or on disk.

The first nice enhancement you will notice is that now there are several single-key commands:

A -	AUTO	L -	LIST
C -	CLS	M -	MERGE
D -	DELETE	N -	NEW
E -	EDIT	P -	LLIST
K -	KILL	S -	SYSTEM

The entry of any of these letters in the proper syntax will be interpreted as if the entire command had been typed.

Another single-key command is 'U' for 'UN-NEW'. This function can restore the resident BASIC program if executed after typing NEW, after exiting to SYSTEM, or even after a system RESET.

In addition, the period keys, the comma keys, and all four arrow keys are utilized to easily list various lines of the program without having to type in any line numbers. For example, typing the left arrow will automatically list the first line of the program, the right arrow the last line, the down arrow the next line, the up arrow the previous line, the period the current line, and the comma automatically puts the current line into edit mode.

Another feature incorporated into SNAPP-II is XREF: the cross reference facility. This enhancement allows you to get a cross reference listing of any and all line numbers and integer constants used in a program. Want to delete line number 1234 but don't know or don't recall whether that line is referenced by a GOTO or a GOSUB? No problem. Type in

'X1234'

and the XREF feature will automatically list all the lines that reference line 1234 or that contain the constant 1234. What if you want to use the variable YY$ but can't remember if it has been used already? Just type in

'XYY'

and you'll get a listing of all lines containing the variables YY, YY%, YY$, YY!D, YY#, as well as any arrays beginning with YY. Each line number listed specifies the variable type. By typing in 'X' to invoke this facility and following it by any one of several possible syntactical combinations, you can get all of the above and more sent to screen or printer.

Whenever an XREF search for a specific line number reference or variable reference results in a listing of the corresponding line numbers, you can press the 'F' and ENTER keys repeatedly and receive a consecutive list of all the lines found in that search, one at a time.

XDUMP, the dynamic variable print facility, is just that . . . DYNAMIC! This routine allows the programmer to investigate the current values of any or all variables referenced anywhere in the program. When the program is exited under normal conditions—whether by 'STOP' or by pressing the break key—the variable values are still in memory. However, XDUMP must be used before editing any lines since editing will automatically clear all variables. The XDUMP module is invoked by typing 'Z' and following it by any one of several possible syntactical combinations. Typing in 'ZA', for example, will list all values of all variables A (A%,A!,A#,A$, and arrays). ("That's impossible! A% can't be 125," the programmer mumbles to no one in particular.) When using this feature, I can pause in the middle of a running program to examine variables and then CONT execution. I have found this invaluable!

Another very powerful and multi-talented addition provided by SNAPP-II is XRENUM, the enhanced renumbering facility. After considering this facility's various capabilities, I found the following to be *most* useful: XRENUM allows the programmer to specify an upper limit to a block of code to be renumbered. It allows blocks of lines to be relocated in the program. And it provides the ability to duplicate portions of the code elsewhere in the program.

XRENUM also corrects some deficien-

cies inherent in Radio Shack's Renumber. Without renumbering, it allows you to search the program for non-existent lines that are referenced and, if you want, it can renumber without altering any references to such lines. This might be necessary if you have referenced routines which have yet to be added to the text.

Moreover, whenever XRENUM is used, you get two tidbits of information: (1) how many lines in the program, and (2) how many bytes the program occupies.

The module is accessed by typing

'R newline, increment, startline, endline'.

Next we have XFIND: the string/keyword cross reference facility. This routine will list references in the code to any or all literals specified. For example, if you use the phrase 'press <ENTER> to continue or Fl to exit' or any other similar type of phrase more than once in a program, you can locate the lines containing this phrase simply by typing

'F"press" '

XFIND will list all line numbers containing 'press'.

In addition, the '@' sign can be used as a wildcard when searching. For example, F" @1 " would, among other things, find all occurrences of space-Flspace and spaceflspace. This same facility will allow you to list the lines containing any or all keywords. For example, if you want to locate every line in which you used the keyword 'OPEN', the command

'F "OPEN" '

will do this for you.

Note that the syntax is slightly different for strings and keywords. The syntax for strings contains a trailing quote while the syntax for keywords does not.

XFIND duplicates one of the abilities of XREF; whenever a search for a specific string or keyword results in a listing of the corresponding line numbers, you can get a one-at-a-time, consecutive list of all the lines found in your search by repeatedly pressing the 'F' and ENTER keys.

XFIND is invoked much the same as XREF and XDUMP: Type in 'F', followed by any one of several syntactical possibilities, and you will get various types of screen or printer output.

The final facility in SNAPP-II is XCOMPRESS, the program optimization facility. This routine is worth its weight in chips! XCOMPRESS can reduce the size of a BASIC program to its absolute minimum. When I need as much string space as possible in RAM, or want to conserve disk space, or just want to fit in a larger running program, this facility provides the means to accomplish such tasks.

XCOMPRESS uses the following operations on the resident program to achieve its goal:

— removal of remarks
— removal of irrelevant blanks
— removal of irrelevant tab characters
— removal of extraneous colons
— removal of the LET keyword
— removal of quote marks at the end of a line
— removal of GOTO following a THEN or an ELSE
— removal of extraneous characters from variable names
— removal of non-executable code
— removal of explicit variable typing characters when a corresponding DEFxxx is present
— merging multiple statements into single lines
— renumbering the program to make line numbers as small as possible.
— removal of specific variable identifiers from NEXT statements

As the compression is taking place, various messages on the screen advise you how many bytes are being eliminated. It is quite possible that a fully compressed program will be 30–40% smaller after XCOMPRESS.

Most of the above phases are optional, which allows you to pick-and-choose those you want to omit and those you want to retain. The sequence of options that I frequently request is to leave remarks, not to renumber, and not to merge lines. This keeps my program in its original format with line numbers that I set up and the remarks that keep me from going crazy trying to figure out what a routine is supposed to do.

Bob Snapp correctly puts in a word of caution that a copy of the original uncompressed version should be saved since a fully compressed program is almost unreadable.

SNAPP-II is currently available for TRSDOS 2.0 and TRSDOS-HD and is being developed for the various Model 16 modes.

Any review of a Snapp-Ware product would be incomplete without mentioning Bob Snapp, his company and customer service. I have used many of the Snapp-Ware products and have had several occasions to call the company directly. I have never found the customer service representatives to be anything but 100% helpful and courteous. They are always willing to answer any questions about their products and equally as willing to take any criticisms. In fact, if Mr. Snapp is there, he usually answers the phone himself and takes care of any questions or problems. When all too many companies today take the attitude: "Well, you bought it—now you figure it out," it is indeed a pleasure to do business with a company like Snapp-Ware.

NEW DOUBLE-SIDED DISK CP/M FOR MODEL 16 (Z80 MODE) FROM P&T

Pickles & Trout has just announced a double-sided version of CP/M 2. It supports either the Model 16 (Z80 Mode) or the Model II with any combination of single and double sided drives, any of which can be Thinline style. This new version is fully compatible with other versions of P&T CP/M 2, and no conversions of programs or data are needed.

Double sided diskettes have a formatted capacity of 1.2 Mbytes.

The price for a new P&T CP/M 2 user is $220. Upgrades of single sided P&T CP/M are $50, and upgrades of double sided P&T CP/M to the Model 16 version are $15. Your original distribution disk must be returned with an upgrade order.

Available from Pickles & Trout, P.O. Box 1206, Goleta, CA 93116.

Hard Disk: When should you convert?

by Richard H. Young

When should you jump on the hard disk bandwagon? Possibly never. However, you should add hard disk right now if you meet both of these conditions:

(1) Your current application is essential and absolutely requires hard disk. For example, you have a large inventory file or accounts receivable file that will not fit on a floppy disk, and

(2) by using the hard disk, you will save more than its cost within no more than two years.

Almost every month there are announcements of significant increases in hard disk capacity and significant reductions in cost. For example, one vendor who advertises in 80 Micro offered a 10 megabyte fixed Winchester last September at $5295. In August, 1982, the same vendor offers a 120 megabyte subsystem at $10,500. This is a reduction from $530 per megabyte to $88 per megabyte (or 83%). A representative of a major hard disk manufacturer told me recently that his company expected to announce a 100 megabyte hard disk at about $6000 by early next year.

What are the advantages of hard disks (vs. floppies)? (1) Large files which would require more than one floppy disk (500,000 bytes on a Model II, 1,250,000 bytes on a Model 16 in native mode) can be accommodated. (2) Your entire library of system and application program files can be available at all times. (3) Programs which perform many file accesses will run much faster (often in 10% to 30% of the time). (4) Operator instructions can be much simpler when floppy disk swaps are eliminated.

Backup procedures for hard disks are not yet very satisfactory. Some Radio Shack personnel recommend that only incremental backups need be taken regularly, while backup of the full hard disk should be done only monthly or quarterly. (In an incremental backup, only those files which have been updated since the last backup are saved.) However, consider the case where

(1) you have five million bytes (running on a Model II with Radio Shack 8.4 megabyte hard disk) in fifty inventory and other data files, some of which are updated at regular intervals varying from daily to quarterly, while other files are updated at arbitrarily changing intervals,

(2) you have one million bytes of other (constant) information on the hard disk,

(3) you take a quarterly full backup and a daily incremental (all on 500k floppy disks), and

(4) you lose the hard disk at the end of the eighth week of the quarterly period.

Since you do not know on exactly what date each file was last updated (because your hard disk directory is gone), the only safe and easy, though time consuming, way to restore from incrementals, is to load the full backup taken at the end of the previous quarter onto the hard disk, then load each of the forty days of incremental backups. However, this method involves loading probably *110 or more* floppy disks and will require many hours to complete.

An alternate method which might considerably reduce the number of disks to be reloaded (but at the expense of more complexity and increased potential for error) is to take a hard-copy listing of the hard disk directory each time an incremental backup is taken, then, upon hard disk failure, (1) restore the full backup from the end of the preceding quarter and (2) restore each file which has been updated since then from the backup disks for the day of its last update. This method is very tedious and may result in disastrous errors if some of the restored files turn out not to be the latest updated versions.

Newer, more convenient, methods of backing up hard disks are regularly appearing. These include streaming tape (tape which constantly moves, as in an audio or video tape recorder), inexpensive (about $5000, with controller) 9-track tape drives, which may include the added benefit of being able to read and write industry standard tapes which are compatible with large computers, and various tape cartridge-based systems.

For just plain data storage, floppy disks are considerably cheaper than hard disks. If you rotate each disk's backup among a total of three disks (and I hope to goodness you do!) and your disks cost about $3 each, data storage under TRSDOS 2.0a will cost about $18/megabyte ($3 x 3)/(500,000/1,000,000). About the best I would hope for in the next year or so for hard disk prices is $60/megabyte (or about three times the cost of floppy storage). However, it is reasonable to expect that hard disk cost per megabyte will converge with floppy disk costs within two to three years.

Another competitor may make both

floppies and hard disks as obsolete as the punched card. My guess is that, in no more than five years, huge solid state memory devices will become available at prices (per megabyte) which are dramatically lower than hard disks. In addition, these devices will be something like *one thousand* times as fast to access as hard disks. Backup devices will probably be inexpensive non-resuable laser-written media (written and read at per-byte speeds which are thousands of times as fast as the fastest commercial magnetic tape drives). A 16-bit microcomputer which has these two features will be able to process any kind of business data hundreds of times as fast as the fastest large business "mainframe" computer today (and probably significantly faster than the large mainframes of tomorrow as well, considering their much longer average circuit length).

If a hard disk will pay for itself within a short time, buy it. Otherwise, stay tuned for amazing developments in storage media in the next few years, but keep on with the old "floppy ballet" until they occur.

Model 16 Owners Get More Attention!

A new publication for the 16 has been announced. *Sixteen*, published by Lawrence Falk, will make its debut about the end of July. The monthly publication will sell for $25 annually in the U.S., $32 in Canada and Mexico.

Mr. Falk tells us that Sixteen will cover topics of general interest to users of the Model 16, including program listings, product reviews, helpful hints, and other matters.

We welcome another viewpoint on the II/16 scene. More information about the Model 16 should be a boon to all users.

Mr. Falk also publishes Rainbow Magazine, a monthly directed at the interests of users of the Radio Shack Color Computer. For more information, contact him at (502) 228-4492, or write to *Sixteen*, P.O. Box 209, Prospect, KY 40059.

Another Dual Processor Enhancement for the Model II

The Radio Shack Model 16 enhancement for the Model II has a new competitor in the 8086 Dual Processor Option (DPO) from Veritas Technology. This board (no other modifications are required, just insert the board) allows you to run the Digital Research CPM-86 operating system *right now*, and we're told that it provides complete file compatibility with existing systems based on Pickles & Trout CP/M. The DPO can also run under Microsoft's MS-DOS.

The Model II will continue to be able to run in its native mode (just like a Model 16 does) using the Z80 chip. The Z80 is used for I/O processing under either 16 bit operating system (just like the Model 16).

This board will run in a Model 16 as well, and will allow its Radio Shack host to use the rapidly developing repertoire of programs for 8086 and 8088 based machines, such as the IBM PC. The board comes with 128K of memory, upgradable to 256K.

We understand that the basic price for the 8086-based DPO is about $1,500, and that a similar product based on the 8088 chip is expected to sell for about $1,000.

For more detailed information, call Veritas Technology, Inc., (408) 263-0646, or write to them at 2375 Zanker Road, San Jose, CA 95131.

Much madness is divinest sense
To a discerning eye;
Much sense the starkest madness.
'Tis the majority
In this, as all, prevails.
Assent, and you are sane;
Demur.—you're straightway dangerous,
And handled with a chain.
—Emily Dickinson
LIFE, XI

"IT'S BROKE" Or WHAT TO DO WHEN YOUR COMPUTER FAILS

by John Thornton
P.O. Box 223
Newtonville, NY 12128
518/381-7123

As with all electronic equipment, sooner or later (hopefully later), something has to go wrong with your computer. In this article, an attempt is made to give you a fundamental set of rules to follow when trying to determine if the machine itself needs repair or if some other factors may be causing a computer malfunction.

Two categories of break-downs exist which you must be concerned with when a computer malfunction occurs: (1) permanent break-down, and (2) intermittent break-down.

Permanent break-down occurs when the computer repeats the problem over and over. This is by far the easiest to locate and repair because the problem continues to exist after the computer gets to the repair center.

Intermittent break-down occurs when the computer is functioning well, then for no discernible reason causes an error, and then resumes functioning well again. This type of failure is a nightmare for both the user and the repair center.

To aid in the diagnosis and the actual repair, the repair center needs several pieces of important information from you. NEVER send your computer to repair with a note reading "It's broke." This practically guarantees that the cause of the failure will be impossible to find.

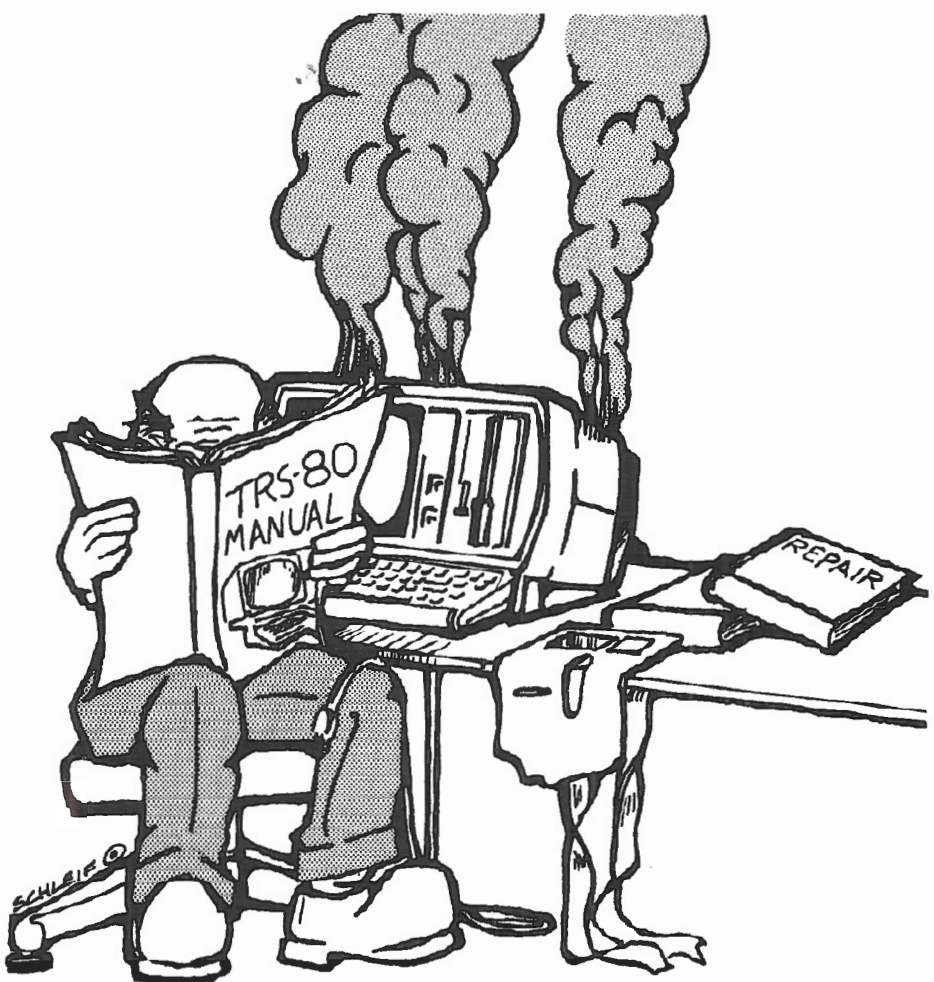
The information that the repair center needs to receive from you in order to provide the best possible service consists of the following:

VITAL INFORMATION

1. Record the time of day the machine failed.
2. If tried again later, did the machine work properly?
3. Were you using new or different hardware or software?
4. What actions did you take? (Keep a record).
5. Note damage to data or programs where possible.

To illustrate the importance of recording this vital information for your repair center, the following incident is a case in point:

I know someone who had a computer that failed almost daily for two months, and he called me in to help locate the problem. After the computer had made several trips to the repair center where nothing could be found wrong, frustrations were mounting and tempers were wearing thin. I suggested that the user keep a failure log near the machine to record the problems every time they happened, which he agreed to do. It



didn't take long while maintaining this log to discover that the machine failed every day at 3:00 PM or very shortly thereafter.

By using this method to track the problems, the user was able to determine that there was nothing wrong with his machine—the problem was being caused by an outside source and this source was, predictably enough, the power company. They had repeatedly denied any power changes in the area to us as we tried and tried to pinpoint the computer's problem. In reality, the power company was switching power in and out to a factory about ten miles down the road from where the computer was located every day at the same time.

After this experience, it is my recommendation that every Model II/16 user have a regulator attached to the power line running into the computer as well as the drive expansion. There is just no way that the computer can handle power switching, and a regulator can help to balance the power flow on a quick switch.

NOTE: Another method to employ when trying to narrow down a problem is to use another computer with the same program disk. If the program works in one computer and not in the other, at least you can be sure it's not a program problem.

REMEMBER: Always try another set of disks since this seems to be the most troublesome area in microcomputing.

Also, make sure that your machine receives periodic preventive maintenance.

Knowing the procedures to follow and then following them may save you a lot of time and frustration when experiencing problems with your computer.

"DISK FULL"

REVIEW OF PRO-80 BASIC

By Bradford Russo
3014 Columbia Avenue
Lancaster, PA 17603
717/397-3461

Enhacements for the Radio Shack BASIC Interpreter

In our inaugural issue, we published an article entitled "Lamentations on the Radio Shack BASIC Interpreter" (May/June, 1982, pp. 2–3), in which nine different limitations of the RS BASIC interpreter, ranging from slow calculation to lack of double precision logs and exponentials, were outlined and discussed.

Should you, too, have felt constrained by the restrictions imposed by your BASIC Interpreter, you are not alone. Such discontent is wide-spread among BASIC users and has been noted and productively acted upon by various software programming groups.

Since the previously stated editorial goals of this magazine are (1) to help you stay current in a rapidly changing market, and (2) to enable you to utilize your micro to its maximum potential, offered here for your consideration is a review of one BASIC Interpreter enhancement package: The Pro-80 Extended BASIC, from Pro-80 Systems in Cedar Falls, Iowa.

We commissioned Bradford Russo of Computer Software Specialists in Lancaster, Pennsylvania, to bench test the Pro-80 and discuss his findings and conclusions with our readers.

VERY VALUABLE FEATURES

Pro-80 BASIC contains a new edit subcommand: OVERWRITE, invoked by the letter "O". This feature combines the best of BASIC's "Change" and "Insert" by replacing one letter at a time. Like the "Insert" command, OVERWRITE can be exited by the Escape key. Unlike "Change", however, the number of characters to be changed need not be specified.

PRO-80 BASIC features the ability to edit line numbers. This makes it possible to edit line numbers and provide instantaneous re-location of code, without the time-consuming process of re-typing the entire line.

Perhaps the most astounding feature, in my mind, is that code can be edited during program execution, *without destroying the variables*! Historically, this was always impossible, as the slightest code change cleared all data storage areas.

Still another valuable feature is a built-in serial printer driver. By using the commands APRINT, BPRINT, ALIST, BLIST, etc., the peripherals attached to the A and B RS232 ports can be operated.

Pro-80 BASIC contains two very good features for subroutines. One is that a subroutine can be called by a label name instead of the beginning line number. The code of the subroutine must have line numbers—as do ALL lines in BASIC—yet with Pro-80 BASIC, the subroutine can be called by its label and still be relocated (or renumbered) to any position.

The other amazing subroutine feature is the MULTILINE :DEF FN statement. Now, the user defined function is no longer merely the evaluation of an expression. With Pro-80 BASIC, it can include any BASIC commands, in any quantity, running to any number of lines of code. The function definition literally becomes a subroutine. In addition, it is totally relocatable and is called only by a label—the function's name.

Pro-80 BASIC contains several useful programming aids: the ability to protect a portion of the screen from scrolling, the ability to write onto the end of a sequential file with OPEN "E" (extend), and full support of PEEK and POKE.
Debugging aids available include single

step execution and a dynamic variable dump that lists the contents of all variables.

FEATURES OF SOME VALUE

With Pro-80 BASIC, it is possible for variable names to be extended to three significant characters. Certainly, three significant characters are better than two, but I find it upsetting that the third must be alpha only. This feature is not really good enough to create English-like descriptions of the data.

Pro-80 BASIC contains numerous string manipulation commands and functions. These commands and functions do not make it possible to perform anything that couldn't be done before, but they do greatly simplify certain operations. One command can accomplish what formerly required several processes, or even a reiterative process.

One of the more interesting of these string manipulation commands is EVAL ( ). BASIC's regular VAL ( ) function simply extracts a numeric value from the front of a string. EVAL ( ) evaluates an arithmetic expression, using all the recognized math operators and functions. It will even accept variable names, evaluating them as a part of the expression.

One of the features of extended BASIC that excited me with its potential was *Program Linking,* where new code can be brought into RAM (from disk) while retaining all variables, pointers, open files, etc. I originally wanted to put this feature in the category with the "Very Valuable Features." However, after careful consideration and analysis, I was forced to include it in the "Of Some Value" category only. Program Linking is intriguing but not essential. Good programming practice dictates that any program whose code will not fit in storage should never have been considered a single functional unit to begin with; it should be several separate programs, each having a similar sub-function.

FEATURES OF NO REAL VALUE

I found that while some of the string functions had some value, Pro-80 BASIC contains *numerous* string functions which serve no practical purpose whatsoever. The manual that accompanies Pro-80 BASIC admits they are included only because they are possible. They make it possible to do, in one command, what would otherwise require a great deal of code. I just don't feel that this is sufficient justification for their inclusion. BASIC already suffers from a lack of standardization between implementations and a lack of mobility between machines. Unnecessary features, such as these Pro-80 BASIC string functions, only worsen the problem.

One command allows the screen to be rolled up or down, one line at a time. When this command is implemented, one line at a time is shifted from top to bottom, or bottom to top. In my opinion, this creates a chaotic, almost unreadable situation on the screen. It destroys our common frame of reference where things occur from top to bottom. Trying to read one of the rolled screens is like walking into a room, only to find yourself standing upside down on the ceiling.

Another example of an almost useless feature is CKKEY. This command checks to see if a key was pressed. The identical thing can be accomplished with BASIC's INKEY$, which not only checks to see if a key was pressed, but also returns the key that was pressed (and CKKEY does not).

BAD FEATURES

Some features of PRO-80 are not only of no value, they are actually destructive. For example, the parentheses formerly required to surround the argument of all functions are now optional. This option may save a few hundred bytes of storage in a large program, but it destroys the readability of the code, and actually produces errors in some cases.

Another truly poor characteristic of Pro-80 BASIC is the ability to manipulate the pointer a sequential file, just like random access. You have no business doing this! It destroys the fundamental concepts and advantages of sequential access. If you are going to do random access, then use a random access file.

THE BUILT-IN SORT

And now, the feature you have all been waiting for . . . a built-in sort!

Pro-80's sort is much faster than any BASIC code, for the simple reason that it is written in machine language. The manual states that a shell metzer algorithm is used that actually swaps the

contents of the variables so they are sorted to the parameters specified.

Here are some benchmark tests:

		Number of elements in array: 100	200	400
# of	2	6.77 secs.	16.08 secs.	38.48 secs
characters	3	10.02	24.84	61.86
in each	4	13.74	34.17	84.34
variable	5	17.85		
	6	21.11		
	8	28.23		

Each block above shows the time it takes—in seconds—to sort the specified array. In addition, each of the times is the average of four different tests. The characters used for these tests are the digits 0–9.

However speedy this sort may be, it is not without its limitations. For example, the sort may only be used on string arrays. Thus, numeric data must be stored as string.

Another limitation: it can only sort one array. Additional arrays with associated data cannot be "carried along." However, it will accept two dimensional arrays.

General Observations

Pro-80 BASIC is composed of a set of several separate machine language modules that are loaded into RAM—in any desired combination—with the BASIC interpreter. These modules consume additional memory, reducing what is available to the user. For example, the sort module consumes an additional 911 bytes, and the extended string functions take 1219 bytes. It is interesting to note that several modules loaded together take slightly less space than the sum of them individually.

Final Conclusion

My final conclusion is that Pro-80 BASIC does what it claims it can do: it extends the capabilities of Radio Shack's BASIC interpreter.

Before you buy, you should be aware that some of the features contain potential hazards, and the sort is not as fast as some I've seen. However, my overall recommendation is that the package is worth its price. After you've bought it, you should feel satisfied that your money was well spent.

Be favorable to bold beginnings.
—Virgil
GEORGICS I

"Why You Should Learn To Back-up Your Disks"
or
"In Computer Jargon, A Grandfather is Not What You Think"

BARBARA'S ADVENTURES IN COMPUTER-LAND

We asked for suggestions from our readers, and boy! did we get suggestions. (See "Letters to the Editor" column.) They came by phone and by mail. They came from California, Texas, Idaho, Montana, Louisiana, New York, Florida, France, Papua New Guinea, and all points in between. They came from doctors, accountants, programmers, consultants, attorneys, and any other profession imaginable. They came from people who work with their micros every day in a business environment and know a great deal. They came from people who want to learn more before they take the plunge to buy microcomputers for themselves.

And they came from the newcomers to the microcomputing business—the novices, the beginners, the ones who are struggling against what seem at this point to be insurmountable obstacles. These courageous people are constantly striving toward, if not complete mastery, at least some measure of control over that inscrutable piece of metal and circuitry—the TRS80 Model II (or 16, as the case may be).

The suggestion most often heard from this group of adventurous souls was: How about a column just for us? How about dropping all the technical jargon and write on a level we can follow? How about an article on a topic that can be understood by someone who doesn't have a Ph.D. in Computer Science? How about some help?!

Dick Young, the publisher of two/sixteen, has been micro-computing for well over a year and has been working with computers in one capacity or another for more than 25 years. He is the recipient of the suggestions that come into this office, and he recognized immediately the need for a series of articles or even a column for the novice. But he has attained a level of proficiency that does not allow him to remember what it's like to be down at the bottom, back at square one, where the mere thought of confronting the computer face-to-face can be traumatic.

The logical solution to this problem was to find a novice, someone who was only learning about computers and who would be willing to write about the slow and tedious and sometimes painful learning experience.

So he turned to me, his intrepid associate editor, dyed-in-the-wool English major who refused—purely on general principles—to take any computer science courses in college. At my alma mater, computer science majors were sneered at by us English majors for their lack of appreciation of the fine arts, just as they sneered at us in return for our refusal to remove our heads from our literature books and face the real world.

This information is offered only for background purposes in order to establish my credentials as an absolutely and totally green novice. I have named this column "Barbara's Adventures in Computer-Land" because I feel a great affinity for Alice these days and know exactly *how she felt when she fell down the rabbit hole into a completely strange and wonderful new environment where excitement, adventure and danger lurks behind every door.*

My introduction to the TRS-80 Model II came about via Wordstar which runs on the CP/M operating system. Dick Young showed me how to insert a disk into Drive 0, handed me the Wordstar manual and said, "Let me know if you have any questions." Several articles had been submitted to the magazine that needed editing and what better way to edit than with Wordstar? Of course, way back then (6 weeks ago), I had no ide at all what I was doing.

However, I do know how to read and Dick told me, "If you can read, you can learn Wordstar." I took a deep breath, opened the manual to page one, and started reading. I practiced and experimented and before too long, I had opened a document file containing my very first article.

What a rush! To be able to add and delete and go front and back within the file and move blocks and exert control over words on the screen in front of me gave me such a sense of power—I loved it! I still love it.

I spent hours and hours typing and experimenting and playing around with the various editing capabilities Wordstar offers. While I was doing my editing job, I was also acquainting myself with word processing. Soon I had five files of my own on one disk, and I felt so proud. I was really a Wordstar whiz.

Until one day . . .

I came to work before Dick—he's th only one who really knows anything about the computers around here—and I though I'd get right to work. I turned on the computer, inserted my disk, and waited for the reassuring clicks and grinds I usually get from the Model II

telling me it's working; instead, I got nothing. Silence. And the screen stayed white instead of turning black. Then an ominous message appeared on the screen: "BOOT ERROR DC."

I said to myself, "This can't be right," so I removed the disk, hit the reset button, and inserted the disk again. The same thing happened: several seconds of silence and then "BOOT ERROR DC." I may be a novice, but I'm not completely stupid so I decided to wait until help arrived.

When Dick came in, I showed him what had happened. His initial reaction was not reassuring. He said, "Oh dear" and began looking around to try and determine what I had done and how much damage my actions had caused.

The source of the Boot error was soon located: *I HAD FAILED TO TURN ON THE POWER TO THE DISK EXPANSION DRIVE WHEN I TURNED ON THE COMPUTER'S MAIN POWER SWITCH*. The Model II, with an added disk expansion drive unit, will not operate or "boot up" unless the drive unit is turned on.

The *TRS-80 Model II Owner's Manual* tells us that error code DC means: "Floppy Disk Controller Error. Defective Diskette—Try Another." That is so true. The person controlling the floppy disk—me—made an error that resulted in the disk becoming defective. Unfortunately for those of us who have tried to power-up the computer without turning on the extra disk drive, our disks may not have been defective when they were inserted, but they were when we took them out.

Luckily, such an error only damages the first one or two files in the directory. It does not rip and slash indiscriminately through the entire disk, wreaking havoc everywhere; it is quite methodical. And because the damage is so easy to locate, it is really quite minimal, no matter how earthshaking it may appear to be at first.

The first thing we did was to run a "DDCHECK" to see if any real damage had been done. We didn't have to go far before this message appeared: "SEEK ERROR, CODE = 0011 0010 BDOS ERROR ON A: BAD SECTOR." Those words can strike terror into the hearts of the brave and strong, so you can imagine what it did to a trembling coward like me.

We found several Bad Sectors on my one and only disk that day. Dick said, "What we can do is erase the damaged portions of the disk and save the rest of your files by copying them onto your back-up disk."

"My what kind of disk?" I said.

He said, "Your back-up disk."

I continued to look blank. He said, "Are you telling me you haven't made any back-up disks?"

And that is how I learned the importance of making back-up disks or *establishing a generation* of disks to protect my work.

Apt words have power to suage
The tumours of a troubled mind.
—Milton
SAMSON AGONISTES

ESTABLISHING A GENERATION

A generation consists of three consecutive disks: a Son, a Father and a Grandfather. The disks were given these names because you work on them in chronological order: youngest to middle-aged to oldest, or Son to Father to Grandfather. The Son (the youngest) is your most current disk or the one you are working on at the present time. When you have added some new material that you wouldn't want to lose in the event of a BOOT ERROR or a BAD SECTOR, it's time to back-up the Son disk, the one that contains your new text.

The back-up process involves nothing more than copying what is on one disk onto another so that you have two copies of the information in the event of damage to one of the disks.

When you are ready to back-up the Son disk, you are going to copy on to the Grandfather disk or the last one in the series. The Father disk, or the one in the middle of the series of three, then becomes the last in the series or the one that will be copied onto next time. (See Figure 1.)

This concept can appear confusing and intimidating until you get the hang of it. When you back-up the Son onto the Grandfather, the Grandfather becomes the Son, the Son becomes the Father and the Father becomes the Grandfather. That's not too bad, but then the next time you go to back-up, the Grandfather/Son becomes the Father, the Father/Grandfather becomes the Son, and the Son/Father becomes the Grandfather.

The only thing more confusing than this is what can happen if you forget which disk is your most current and try to straighten them out by looking at the directory of each one and trying to remember what work you had done most recently. (That's happened to me, too.)

BACKING-UP

The back-up procedure is quite simple once you've figured out which disk you're going to copy from and which disk you're going to copy on to:

1.) Put the Son disk into Drive 0 (A) and the Grandfather disk into Drive 1 (B)

2.) Erase the contents of the Grandfather in Drive 1 (B) by entering

A>ERA B:*.*

This is an important step in the back-up process; you do this to avoid resurrecting old files which you have deleted on this disk in the past.

The computer will ask you to verify your decision to erase everything. It will ask: A> All (Y/N)? Be assertive—don't let the computer question your decision! Enter "y", and you should see the light on Drive 1 (B) go on as the computer erases the Grandfather.

3.) When the Grandfather is empty, the light on Drive B will go out and the light on drive A will come back on. The screen will show you that Drive A is waiting for your next command: >A.

4.) To copy the contents of the disk in Drive 0 (A) onto the now empty disk in Drive 1 (B), do a "FASTCOPY"—a Pickles & Trout CP/M command.

Once the FASTCOPY command has been implemented, it will prompt you through the steps needed to complete the procedure.

When you type in: A> FASTCOPY

You will receive the response:
"Fastcopy routine ver 2.24 Pickles & Trout"
"This program copies all files from the source disk to the destination disk while maintaining all file attributes."
"Enter source drive (A, B, C, D, <Break> to quit):"

You should type in: A

"Enter destination drive (A, B, C, D, <Break> to quit):"

You should type in: B

"Once the copy has been started, you may abort it by typing <break>. Mount source disk on drive A and destination on drive B (hit <enter> when ready)."

You should be ready, so hit "Enter" and then sit back and watch.

First the Drive A system light will light up as FASTCOPY reads the Son disk. The screen will show a read-out that looks something like this:

```
Reading ED.COM R/W, DIR,  user = 0
Reading ASM.COM R/W, DIR, user = 0
```

Then the Drive A system light will go out and the Drive B system light will come on as FASTCOPY copies onto your Grandfather disk what it has just read on the Son disk. The screen will look like this:

```
Writing ED.COM R/W, DIR,  user = 0
Writing ASM.COM R/W, DIR, user = 0
```

The entire FASTCOPY process takes about five minutes, depending on how full your disk is. Five minutes doesn't sound like a long time, but when you're just sitting there, watching the screen, and waiting for the computer to finish up so you can work some more, five minutes *is* a long time. So get up and walk around . . . take a coffee break, whatever. The computer will finish the FASTCOPY without any further help from you.

A question I had concerning this back-up process is how often should it be done? Every hour? Every night? Every week? In the tradition of wise men throughout history, Dick answered my question with a story.

He knew a dentist who often worked with children, teaching them the importance of good dental hygiene. One of the greater mysteries of teeth care was the idea of flossing. Most of the children were familiar with brushing, but flossing was new and different. Among the many questions the dentist was asked was this: "How many teeth do we have to floss? Do we have to floss all of them?" And Dick's dentist friend answered, "No you don't have to floss all of them—ONLY THE ONES YOU WANT TO KEEP."

And so it is with your disks. You don't have to back them up often if you aren't interested in keeping the text you went to the time and trouble to enter. The recommendation from this end is to do it every other day, but that is merely a recommendation. If you enter two files filled with irreplacable information in one day, you probably will want to insure the safety of this information by backing it up onto another disk, whether it is the "scheduled back-up day" or not.

One further note on organization: After you have gone to the time and effort to back-up your disks and have acquired the feeling of security such duplication affords, you must devise some type of labeling system for your disks while they are in storage. Here at the **two/sixteen** office, we use a series of three consecutive numbers, i.e., 90, 91, 92. However, this method is not engraved in

stone; you can use whatever method you want. The important thing to remember is that the number (or name) of your working disk is going to change from day to day as you continue to back it up. So chances are, you don't want to be working with disks that aren't sequentially related somehow.

I was told *absolutely not* to change the numbers on the disks to accommodate my failing memory. I was going around changing the numbers on my disks every other day so that the lowest number was always the first one in the series. WRONG. The three disks are merely rotated from Son to Grandfather to Father and the numbers *do not change!*

I have worked out a solution to end the confusion associated with this rotating disk problem by labeling each of my disks with my name (in felt tip pen only, of course!). I also have a special place for my generation of disks that is separate from the other disks used regularly around here. But the most important thing is that I have been doing it for two months now. It is no longer totally new and strange, and I am starting to feel comfortable around the equipment and handling the disks. Time has a way of smoothing out all the rough edges, and my case is no exception.

Classified Advertising

A non-dealer subscriber may place one ad per issue at the rate of $7 per column inch.

A dealer may place one classified ad per issue at the rate of $20 per column inch. Dealer classifieds must be for the sale of specific products or services at specified prices.

Ads must be for hardware, software, or services which would be of interest to Model II or 16 users (no mystical societies, chain letters, book clubs, etc.).

Ads will be set in the same size type as the text in the magazine.

Closing date for classified ads in the 15th day of the month prior to issue date (e.g., the September-October classifieds will close August 15).

We reserve the right to determine what heading an ad should be placed under (e.g., HARDWARE, SOFTWARE, MODEMS, PROGRAMMING), but we will accept advertisers' reasonable requests for specific headings.

...a quick personal note from

Wayne Green W2NSD/1

WAYNE GREEN, INC.
PETERBOROUGH, NEW HAMPSHIRE 03458

Editor-Publisher

73 Amateur Radio Journal
Microcomputing
80 Micro
Desktop Computing
Selling Micros
Instant Software
Lund 80
Wayne Green Books
80 Encyclopedia
Wayne Green International
(603) 924-0471

11 July 82

Dear Richard, -

Well, you didn't ask directly, but you did get me on your list for an advertising letter, so I'll volunteer.

Having started several magazines...and with more on the way...I have a fair understanding of the economics involved. On that foundation I see a couple of problems. Firstly, it's possible that you have exceeded the price point for even a very specialized magazine with a $30 price for six issues. That's steep even for a big fat magazine.

Then ad costs. Hmmm. At about $187 per M you're a bit high to gather in advertisers. 80-Micro runs today about $11/M, though admittedly it does not have 100% Model II/XVI owners. With 50,000 Model IIs out there, you should be able to beef up that circulation and thus your ad rates.

This is a particularly difficult time to get a new publication started, as I'm sure you know. Three major magazines are up for sale, according to the grapevine and they don't offer to sell when things are going well.

If you get around to some of the shows you might budget some time to stop in up here for a visit...and free lunch.

Regards.....Wayne

two/sixteen magazine

130 East Orange Street
Lancaster Pennsylvania 17602
(717) 392-4861

July 28, 1982

Wayne Green
Wayne Green, Inc.
Peterborough, NH 03458

Dear Wayne,

Thanks for the (I guess) encouraging words.

Subscription price doesn't seem to be a problem. At least, no subscriber or potential subscriber has suggested that it is. You must realize that our average customer has more than $10,000 invested in hardware and software ... and nowhere to turn for help. Most seem to feel that our magazine is worth another $30 per year, even if only one or two personally useful items are obtained. We hope that any reader will find many more than that.

By the way, we also will share such knowledge as we have about their II/16 problems with any readers who write or call. Big fat magazines can't afford to do this, and they can't possess very much specific knowledge about each of the wide variety of computers they cover.

Advertisers don't seem put off by our rates. Of course, we're giving substantial discounts from the rate card. Also, advertisers know that every reader is a potential purchaser of their product. In fact, if you adjust 80 Micro's rates for the proportion of II and 16 readers, its cost per thousand is probably well above ours. We haven't garnered very much advertising, but then we have given priority to the editorial side. The only advertising promotions we have made were two mailings (one in May to 70 potential advertisers and another in July to 120). No phone calls. I expect to pull thirty ad pages by the Nov-Dec issue. And, frankly, that's about all I want, since I expect that fifty pages of (good) non-ad material is about the best we can do (and I'd like to stay below 40% advertising).

We've had very gratifying response from readers who are willing (and even anxious) to write for two/sixteen. Some of them are also regular contributors to 80 Micro, but have not always found their Model II articles welcome.

You see, we're practically in a different business than 80 Micro. As you've told me, 80 Micro is based on volume and must cater to the hobbyist market. Its difficult to be many things to many audiences, so you've understandably had to give short shrift to the relatively small business, professional, and scientific Model II market. However, you must be given due credit for some increase in Model II coverage this year.

Assuming that the three magazines rumored to be for sale are microcomputer magazines, I'm not at all surprised. There are perhaps fifteen large general purpose magazines, like Microcomputing, Byte, and Popular Computing. Many users have subscribed to several. But now we hear from many readers who are drastically reducing the number of general purpose computer magazines they read.

It seems likely that the number of specialty magazines will increase, because they fulfill specific needs which general magazines cannot address. As you probably know, we now have specific competition in the form of Sixteen magazine, which was announced in the August 80 Micro. Frankly, I encourage our own readers to subscribe to Sixteen as well as two/sixteen. There is no limit to the amount of information which can be published about machines as versatile as the Model II and Model 16. In fact, the appearance of a competitor in this specialty field may actually serve to increase, rather than decrease, our own circulation.

When Model II or 16 owners ask me what other magazines I personally recommend, I list InfoWorld, Datacast, 80 Micro (its worth it for the ads, if not the II/16 coverage), and Sixteen (though with reservations until we see the first issue).

Because of all the work that needs doing here, I don't expect to get to any more shows this year. However, if you are in our area on your frequent travels please stop by for lunch or dinner on us.

Sincerely,

Richard H. Young

PROGRAMMING THE MC6845 VIDEO CONTROLLER CHIP

By Marshall DeBerry and Randy LaPorte
3329 Southgate Drive, Apt. 12
Alexandria, VA 22306
703/768-5412

Two programmers from Alexandria, Virginia, had their attention captured and their curiosity aroused by the Model II Technical Reference Manual's description of the MC6845—the Video Controller Chip. Marshall DeBerry and Randy LaPorte were so impressed by the chip's capabilities and its potential for reconfiguring the Model II's video screen that they began experimenting with different video display formats. We are pleased to be able to reveal their test findings so you, too, can produce the same results.

HOW A VIDEO MONITOR OPERATES

The video monitor of the TRS-80 II/16 operates much the same as a television set: a picture or image is created on the screen when an electron beam, produced in the "back" of the picture tube, strikes the fluorescent-coated "front" or viewing area of the screen, causing it to glow. The picture or image is produced by controlling how long and where the beam strikes the fluorescent coating.

For TV sets in the U.S., the horizontal sweep of the beam is usually 15.75 kHz, and the vertical sweep is 60 Hz. If these sweeps were not controlled, the video screen would simply glow and be of little use to the viewer. However, by taking advantage of the beam's speed and the human eye's slow reaction time, the electron beam can be controlled so that it will appear to produce characters or images. Such controls are imposed by circuitry which causes the electron beam to trace a pattern on the screen, like this:

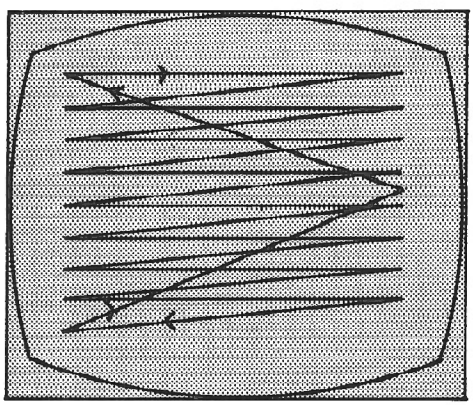


Figure 1

Such a rake-like pattern is still not very useful to the viewer, but if the beam can be turned off for a short period of time, the following pattern can be produced:

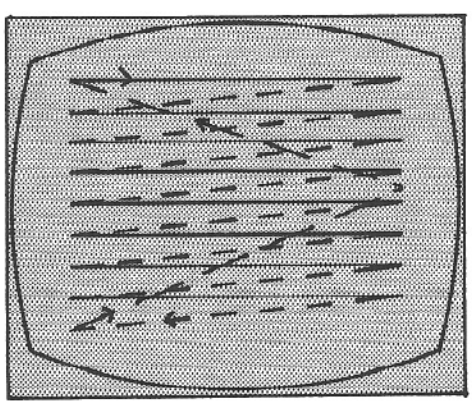


Figure 2

The dashed lines in Fig. 2 indicate horizontal retrace lines, or those areas where the electron beam has been turned off.

As the name "horizontal retrace line" implies, the retrace part of the electron beam scan moves horizontally from one line to another. At the bottom line of the picture tube, the beam makes a "zag" all the way back to the top and starts over again. The beam also has a vertical scan line with a corresponding retrace line.

This information is very interesting, but how can such an arrangement of electron beams and horizontal/vertical retrace lines produce character lines?

The horizontal sweep is 15.75 kHz, and the vertical sweep is 60 Hz. When 15750 is divided by 60, the result is 262.5 horizontal trace lines. However, the retrace lines must also be taken into consideration, and they usually run to 21 or 22 scan lines. These arithmetical computations reveal that there are 240 scan lines available for data display.

The number of characters to be displayed on the screen is also a function of the number of scan lines. The objective now is to increase the number of displayable characters by increasing the number of scan lines. And this objective is achieved by using the INTERLACE SCANNING method. Interlace Scanning represents both the appropriate means for controlling the electron beam and its scan lines and for displaying characters on the screen.

Ordinarily, when the electron beam reaches the end of the screen, it retraces back to the top left corner. However, during the Interlace Scanning procedure, the beam goes to the middle of the top line where it is offset by half a line vertically. The resulting line is traced

between the first two scan lines, then the second two, and so on.

This process can be illustrated as follows (NOTE: MODEL II VIDEO IS NOT PROGRAMMED TO OPERATE IN THIS MODE):

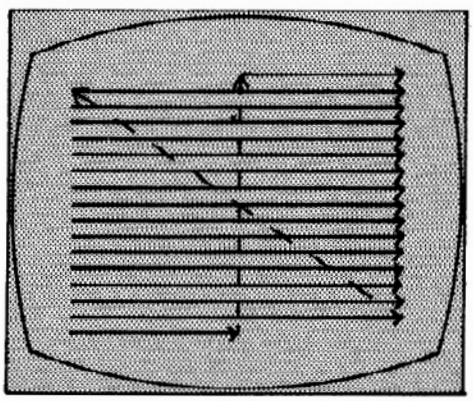


Figure 3

Of critical importance to the display of characters on the screen is timing; the sweep of the beam must be timed *EXACTLY* or the display image will appear totally garbled. Here the computer's system clock has the opportunity to prove its value. Although this clock has many functions, keeping track of the video timing is one its most important.

Another vital component in the process of getting characters displayed on the screen is the special CHARACTER GENERATOR ROM CHIP that stores bit patterns of the computer's character set. These patterns are used by the video circuitry to produce characters on the screen. To create the letter "A", for example, the character generator rom chip will instruct the electron beam to turn on the individual dots on the fluorescent coating and keep them on so that the character can be seen on the screen.

Thus far in our attempt to trace the events leading up to the actual creation of characters on the video screen, we have been shown the importance of the character generator rom chip, the computer's system clock, interlace scanning, retrace lines, etc. The way in which these individual parts act in combination with each other is the way in which characters are produced on the screen. Let's look at this procedure a little more closely:

When the frequency of the system clock (12.48 MHz) is divided by 8 (the number of dots per character width), the answer is the frequency of the character rate clock (1.56 MHz). (The above answer results from the assumption that the 80 character video mode is being used; for 40 character mode, the timing will be different. By setting either the 40 or 80 character mode, a different character clock value is called into use. This value in turn changes the frequency with which the dots are sent to the screen.)

Once these times are generated, they are used to "shift out" the desired characters from the Character Generator Rom chip onto the video screen. For an 80 character line, the dot patterns for the characters to be displayed must be shifted out on a scan line consecutively.

Assume each character in the Rom Chip is made up in an 8x10 matrix of 0's and 1's. It will take 7 horizontal scan lines to represent an 80 character row of letters and numbers. (There is one blank line at the top of the character and two blank lines at the bottom.)

The bit patterns of the characters are used to control the beam so that it appears to the human eye as a fully formed character. In reality, however, the beam is continually sweeping the screen, but at such a speed that the characters being displayed do not have time for their glow to dissipate.

There is yet another part of the picture-making process which must be taken into consideration: VIDEO MEMORY.

When a letter is typed on the keyboard, a code is sent to the various circuitry, requesting the display of that particular character. These request codes are stored in the screen memory area before being sent to the character clock where they are shifted out to the video screen. On the Model II with its 80 characters per line and 24 displayable rows of characters, 1920 bytes of screen memory are needed.

The Model II has a 2K set of memory for video display that is a separate memory area, not part of the regular RAM of the computer. This explains why the video memory area of the Model II is not directly addressable as it is in the Model I and III.

(The foregoing information was provided as background only so that the procedure for displaying characters on the screen could be illustrated. The discussion has been kept as elementary as possible. For more detailed information concerning the workings of video controllers, please consult *The CRT Controller Handbook* published by Osborne/McGraw-Hill.)

THE MC6845

The MC6845 is a 40 pin chip with 18 parameter registers, plus one register-select register. Several of these 18 parameter registers may be regarded as system or video monitor specific; i.e., they are initialized when the system is turned on and are usually not changed by the operating system while it is running.

The Motorola Product Specification sheet for the MC6845 (included in the Model II Technical Reference Manual), shows the following "system" registers for the Model II:

- R0 - Controls the number of characters displayed on each line plus retrace. The number programmed into the register is one less than the actual value used by the video controller.
- R2 - This number represents the area where the horizontal signal (HSYNCH) makes its negative to positive transition. For more details, see the *CRTC Handbook*.
- R3 - The duration of the HSYNCH pulse.
- R4 - Total number of vertical scan lines in a frame plus retrace. Again, this number is one less than the actual value used by the chip.
- R5 - Contains a number used to adjust the vertical scan (VSYNCH) frequency to get close to 60 Hz. This is used to "fine tune" the video monitor so that the electron beam will function within tolerance.
- R7 - A number representing the area where the VSYNCH makes its negative to positive transition. Again, for more details, see the *CRTC Handbook*.

The remaining registers and their uses are as follows:

- R1 - A value representing the number of characters to be displayed on a line.
- R6 - The number of character rows to be displayed.
- R8 - Type of scan mode, which is set by the appropriate two bit code. For example, either a 00 or 10 produce normal video, 01

will produce an interlace synch mode, and 11 will produce interlace synch plus video mode.

R9 - Number of scan lines per character row. The value programmed into the register is one less than the actual value used by the chip.

R10 AND R11 - These two registers control the type of cursor to be displayed. R10 is the cursor start register, and R11 is the cursor stop register. If R10 and R11 contain different values, a block cursor is produced. In addition, bits 6 and 5 of R10 control the type of cursor to be produced, as shown by the following table:

00 - no blinking cursor
01 - no cursor
10 - blink rate: 1/16 field rate
11 - blink rate: 1/32 field rate

To further elaborate on the use of these two registers, assume the following: We are in normal synch mode, 80 characters per line, and R10 contains the value 65H (101D) and R11 has the value 09H (9D). There are 10 scan lines per character. The following character and cursor will be produced:

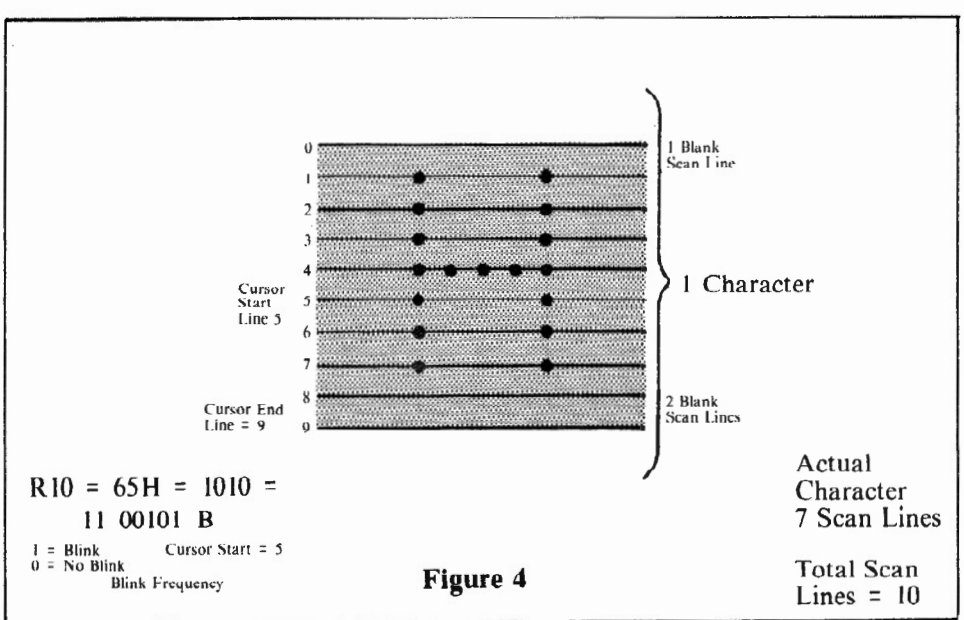


Figure 4

R12 AND R13 - These two registers make up a 14 bit top of page register, which contains the screen memory address for the first character to be displayed. By changing the contents, scrolling can be performed.

R14 AND R15 - These registers contain a 14 bit cursor position value. Movement of the cursor is done by loading these two registers with different values. For example, if R14 has the value of 03H, and R15 has E9H, the combined value is 03E9H, of 1001D. This value will cause the cursor to be positioned at location 1001 on the screen.

R16 AND R17 - These registers comprise a 14 bit value for use if a light pen is attached to the computer. The registers will contain the value of where the light pen signal was detected on the screen.

Now that we know the values put in the registers and how the chip makes use of them, we can write a few simple programs to manipulate the video screen. First, assume that the CRTC contains the following values:

(Values are in Hex)

R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
63	50	55	08	19	00	18	18	00	09	65	09	00	00	03	E9

The BASIC program shown in Listing 1 (See Figure A.) allows us to manipulate the CRTC registers. To use the program, note that two Assembler Programs must be loaded into memory, VPOKE/CMD (Assembler Listing 1) (See Figure B.) and CRTCADJ/CMD (Assembler Listing 2) (See Figure C.). VPOKE/CMD allows us to emulate the POKE function that is not present in MOD II Basic, so that we can easily store data in memory for access by the CRTCADJ/CMD program.

The VPOKE/CMD program is based on a program by Tom Yager that was originally published in the June 1981 issue of *80 Microcomputing*. It has been modified for easier use by our programs.

Note that the value used for the memory storage area in BASIC program one is −3106. This value is a result of the way integer numbers are stored in 8 bit machines. It takes two bytes to represent integer numbers on the MOD II; therefore, the range of integer values that can be stored lies between −32768 and +32767.

In order to correctly place data in memory locations greater than 32767, we use the following formula:

* (65536 − desired address) = address

substituting

* (65536 − 62430) = −3106

Let's illustrate the use of the BASIC program in Listing 1 with a few examples.

To produce a rapidly blinking underline cursor, enter the following values (in decimal):

```
NUMBER OF REGISTERS
    TO CHANGE? 2
ENTER REGISTER #,
    NEW VALUE? 10, 73
ENTER REGISTER #,
    NEW VALUE? 11, 9
```

Execute a CLS to return the cursor to normal.

To position the cursor in the upper left corner of the screen:

```
NUMBER OF REGISTERS
    TO CHANGE? 2
ENTER REGISTER #,
    NEW VALUE? 14, 0
ENTER REGISTER #,
    NEW VALUE? 15, 0
```

The cursor should briefly appear in the upper left corner of the screen before reappearing next to the ANY MORE? prompt.

INTERLACE SYNCH PLUS VIDEO MODE

Now for something a bit more dramatic. Recall the description of the Interlace Scan Mode. It's possible to program the CRTC to enter a mode called INTER-

LACE SYNCH PLUS VIDEO MODE. This appears to be a "pseudo" high resolution mode. To see what it looks like, enter the following values:

NUMBER OF REGISTERS TO CHANGE? 1
ENTER REGISTER #, NEW VALUE? 8, 3

The screen should now have very tiny characters on it. To get back into normal video, reply Y to the ANY MORE? prompt and enter 0 into register 8.

Other configurations of the video screen are possible, as long as careful thought is given to what registers to change and figuring out the bit patterns for the decimal values.

The BASIC program shown in Listing 2 will allow us to "poke" a character directly on the video screen. Earlier we stated that the video memory is contained in a separate 2K area and therefore was not accessible for direct use by the programmer as for the MOD I and MOD III. However, this BASIC program will gain access to this 2K area and allow us to manipulate it. Note that two assembler programs VPOKE/CMD (Assembler Listing 1) and VIDPOKE/CMD (Assembler Listing 3) (See Figure D.) must be loaded into memory before running the program. The BASIC program will prompt the user for the character to be displayed, where to put it, and Presto! there it is.

Notice that the SVC VIDRAM (SVC 94) is not used. When VIDRAM is used, a portion of RAM for the video is set aside, resulting in a loss of 2K of user RAM. Our program directly manipulates the video memory area, not user RAM.

IMPORTANT NOTE!!! If at any time a high pitched whine is heard while running these programs, IMMEDIATELY RESET THE COMPUTER!!! This means either the program is keyed in wrong, or the wrong value was entered into one of the 6845 registers. In any event, damage to either the computer or monitor may result. So please, be careful when using these programs.

We hope you will find these programs interesting and useful. However, future Radio Shack releases of TRSDOS may change the port addresses for the video controller, or the new high resolution graphics package may also change some ports. The programs described here would have to be modified, or perhaps completely re-written.

In the meantime, be creative and enjoy!


```
10 REM  ****  CRTCADJ2/BAS  *****
20 REM  WHEN RUNNING THIS PROGRAM, "CRTCADJ/CMD" AND "VPOKE/CMD" MUST BE
30 REM  LOADED BEFORE LOADING BASIC.  BASIC MUST BE CALLED WITH -M:61440
40 REM  TO PROTECT TOP OF MEMORY
50 DEFINTA-Z
60 DIMV(32)      'Up to 16 data registers
70 DEFUSR7=&HF030
80 DEFUSR9=&HF3B0
90 INPUT"HOW MANY DATA REGISTERS TO BE CHANGED";V(0)
100 FOR I=2TO(2*V(0)) STEP2
110   INPUT"INPUT:  'REGISTER # , NEW VALUE'";V(I-1),V(I)
120 NEXTI
130 FORI=0TO2*V(0)
140 Y1=-3106+I       'Memory address where data will be stored
150 A$=MKI$(Y1)      'Convert memory address to type string
160 A$=A$+CHR$(V(I))  'Tack on the data to the memory address
170 A$=USR7(A$)        'Put the data and the address into memory
180 NEXTI
190 X=USR9(0)         'Call the assembler program to manipulate the CRTC
200 PRINT:PRINT
210 INPUT"ANY MORE";Q$
220 IF LEFT$(Q$,1)="Y"THEN PRINT:PRINT:GOTO90
230 END
```

Listing 1

```
10 REM  ***  VIDPOKE3/BAS  ****
20 REM  THIS IS A  PROGRAM TO ATTEMPT POKING CHARACTERS DIRECTLY ONTO THE
30 REM  SCREEN. NOTE: VIDPOKE/CMD & VPOKE/CMD MUST BE LOADED INTO MEMORY FIRST
40 REM  BASIC MUST BE LOADED NEXT SETTING TOP OF MEMORY TO 61440 (-M:61440).
50 DEFINT A-Z
60 DEFUSR8=&HF3B0
70 DEFUSR7=&HF030
80 INPUT"ENTER THE CHARACTER TO BE DISPLAYED ON SCREEN";A$
90 INPUT"ENTER LOCATION TO BE DISPLAYED AT (0-1919)";B
100 A=ASC(A$)
110 B1=B/256   'Obtains the MSB of screen location
120 B0=B MOD 256  'Obtains the LSB of screen location
130 G=-3104
140 GOSUB 220  'Store data to display on screen at mem location G
150 G=G-1
160 A=B1
170 GOSUB 220 'Store MSB of memory address where data will appear on screen
180 G=G-1
190 A=B0
200 GOSUB 220 'Store LSB of memory address where data will appear on screen
210 GOTO 260
220 F$=MKI$(G)      'Convert address to type string
230 F$=F$+CHR$(A)   'Tack data to pass onto address
240 F$=USR7(F$)     'Put address and data into memory
250 RETURN
260 CLS
270 Y=USR8(0)      'Call assembler program to poke data into screen memory
280 INPUT"AGAIN";Q$
290 IF LEFT$(Q$,1)="Y" THEN 80ELSE END
```

Listing 2

```
      MACRO-80 3.4    21-Nov-80       PAGE    1


                        00100               ENTRY   VPOKE
0000'                   00200               ASEG
                        00300               ORG     0F030H
                        00400 ;
                        00500 ;To understand how this routine works, refer to the description
                        00600 ;of the USR call in the TRSDOS Reference Manual.  Note that we
                        00700 ;pass both the address and the data together as one three byte
                        00800 ;string.
                        00900 ;
F030    EB              01000 VPOKE:  EX      DE,HL   ;Puts addr of len & loc of string in HL
F031    23              01100         INC     HL      ;Go past the string len byte HL pts at
F032    5E              01200         LD      E,(HL)  ;Load LSB of addr of string
F033    23              01300         INC     HL      ;Bump to MSB addr of string
F034    56              01400         LD      D,(HL)  ;Load MSB of addr of string
F035    EB              01500         EX      DE,HL   ;Put addr of string in HL
F036    5E              01600         LD      E,(HL)  ;Load first byte of string in E
F037    23              01700         INC     HL      ;Bump to second byte of string
F038    56              01800         LD      D,(HL)  ;Load second byte of string in D
F039    23              01900         INC     HL      ;Bump to third byte of string
F03A    7E              02000         LD      A,(HL)  ;Load third byte of string into A
F03B    12              02100         LD      (DE),A  ;Put data into memory
F03C    C9              02200         RET             ;Return to Basic
                        02300         END     VPOKE
```

Assembler Listing 1

```
      MACRO-80 3.4    21-Nov-80       PAGE    1


                        00100               ENTRY   GOHERE
0000'                   00200               ASEG
                        00300               ORG     0F3B0H
                        00400 ;
                        00500 ; This routine allows one to directly manipulate the CRTC registers
                        00600 ; of the MC6845 chip.  Refer to the Technical Reference Manual for
                        00700 ; details on the MC6845.
                        00800 ;
F3B0    21 F3DE         00900   GOHERE: LD      HL,0F3DEH ;Beg. loc of data from BASIC
F3B3    0E FD           01000           LD      C,0FDH    ;Port to CRTC data registers
F3B5    46              01100           LD      B,(HL)    ;Number of registers to be changed
F3B6    CB 00           01200           RLC     B         ; x 2 = no. of values passed from BASIC
F3B8    23              01300           INC     HL        ;Point to next value
F3B9    0D              01400   LOOP:   DEC     C         ;Port to CRTC address register
F3BA    ED A3           01500           OUTI              ;Send reg # selected to CRTC addr reg
F3BC    0C              01600           INC     C         ;Port to CRTC data registers
F3BD    ED A3           01700           OUTI              ;Send data to selected CRTC register
F3BF    20 F8           01800           JR      NZ,LOOP
F3C1    C9              01900           RET
                        02000           END     GOHERE
```

Assembler Listing 2

```
      MACRO-80 3.4    21-Nov-80       PAGE    1


                        00100               ENTRY   START
0000'                   00200               ASEG
                        00300               ORG     0F3B0H
                        00400 ;
                        00500 ; This routine will allow one to 'poke' characters directly into
                        00600 ; the Video memory.  Refer to the Technical Reference Manual for
                        00700 ; details on the CRTC chip and video control.
                        00800 ;
F3B0    21 F3DE         00900 START:    LD      HL,0F3DEH   ;Loc of values passed from BASIC
F3B3    5E              01000           LD      E,(HL)      ;Store destination addr in DE
F3B4    23              01100           INC     HL          ;Pts HL to MSB of destination addr
F3B5    7E              01200           LD      A,(HL)      ;Gets MSB of destination addr
F3B6    C6 F8           01300           ADD     A,0F8H      ;To add F800H for proper destination
F3B8    57              01400           LD      D,A         ;DE now has proper addr of VIDMEM
F3B9    23              01500           INC     HL          ;Now pointing to character code
F3BA    3E E1           01600           LD      A,11100001B ;Code to bank select VIDMEM, etc
F3BC    06 21           01700           LD      B,00100001B ;Code to deselect VIDMEM
F3BE    0E FF           01800           LD      C,0FFH      ;Port
F3C0    ED 79           01900           OUT     (C),A       ;Do it
F3C2    ED A0           02000           LDI
F3C4    0C              02100           INC     C           ;'cause C was DEC'ed by LDI
F3C5    ED 41           02200           OUT     (C),B       ;To pass control of VIDMEM back to CRTC
F3C7    C9              02300           RET
                        02400           END     START
```

Assembler Listing 3

READER SURVEY, MAY-JUNE, PRELIMINARY RESULTS

So far, 75 replies have been received to our May-June reader survey asking about hardware and software owned and planned for future purchase. There has not been time to fully evaluate the results, especially with respect to ratings. However, here is a profile of the hardware owned by respondents:

Counts are the total number of units owned (rather than number of owners). Prospective purchases in 1982 are included. All equipment owned by those who own or intend to purchase a Model 16 are included in the Model 16 column. The Ratio column shows the ratio of individual components to the total number of computers, unless marked with an asterisk (*), in which case it is the ratio to Model II computers only.

Item	Model 16	Model II	Total	Ratio
Model II, 64k	10	65	75	84.2
Model 16, 2 drive:				
128k	12		12	13.5
256k	2		2	2.3
Total computers	24	65	89	100.0
Model II to 16 upgrade	1	4	5	6.7*
External floppy disks:				
RS 1 drive	3	17	20	26.7*
RS 2 drive	1	14	15	20.0*
RS 3 drive	1	17	18	24.0*
Siemens 1 drive		1	1	1.3*
Siemens 3 drive		1	1	1.3*
Lobo 2 drive		1	1	1.3*
UR Data 1 drive		1	1	1.3*
"Home grown" 1 drive		1	1	1.3*
Hard disk:				
Radio Shack—primary	2	11	13	14.6
Radio Shack—secondary		1	1	1.1
Corvus		3	3	3.4
Printers:				
LP III	4	14	18	20.2
LP IV		1	1	1.1
LP V	4	11	15	16.9
LP VI	1	11	12	13.5
LP VIII		3	3	3.4
DW II	6	22	28	31.5
Sheet feeder for DW II		1	1	1.1
Epson MX-80	3	4	7	7.9
MX-100	1	1	2	2.3
TI-810	1	2	3	3.4
AJ		3	3	3.4
NEC		3	3	3.4
Centronics		2	2	2.3
Heath H-14	1	1	2	2.3
Qume		2	2	2.3
Other (1)	2	9	11	12.4

Modems:				
RS 1	1	9	10	11.2
RS II	2	12	14	15.7
Novation CAT		4	4	4.5
Hayes SmartModem	1	1	2	2.3
Other (2)	2	6	8	9.0
Terminals:				
DT-1	1		1	1.1
Regent 60		1	1	1.1
Other devices:				
Pickles & Trout Clock Board		5	5	5.6
Printer buffer		2	2	2.3
RS Multi-Pen Plotter		2	2	2.3
Houston Instr. Plotter DMP-6		1	1	1.1
Houston Instr. Digitizer DT-11C		1	1	1.1
Graphics board	1		1	1.1
Mayday Power Supply		1	1	1.1
RS Line Filter		1	1	1.1

Notes:

(1) Includes one each of the following: IDS Paper Tiger, Olympia ES 101 typewriter, C. Itoh F10, IDS 440, MPI 88T, "plotter-printer," OKI Microline 82, OKI Microline 82A, DataProducts M200, DEC LA-120, DEC LA34.

(2) Includes one each of the following: Racal-Vadic VA3412, The Connection, Livermore Star, Telsat 440, Omnitech 710, Bell 300/1200 baud, Micom Micro 800, Racal-Milgo MIC II.

Comments:

All of the reported Model IIs were 64k (no 32k). All of the reported Model 16s contained two drives (no single drives).

About 20% of Model IIs did not have expansion drives, and these are about evenly divided between one, two, and three expansion drives.

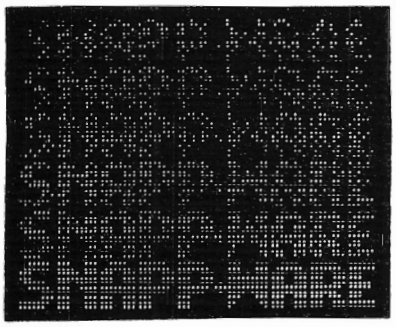
More Model II owners are buying Model 16s than are upgrading.

About one owner in six either has a hard disk or expects to buy one this year.

The Daisy Wheel II is by far the most frequently owned printer.

Your system was liable to periodical convulsions, overwhelming alike the wise and unwise . . .
—Edward Bellamy
LOOKING BACKWARD

. . . take man's vaunted power of calculation. Have we not engines which can do all manner of sums more quickly and correctly than we can? . . . In fact, wherever precision is required man flies to the machine at once, as far preferable to himself. Our sum-engines never drop a figure . . .; the machine is brisk and active, when the man is weary; it is clear-headed and collected, when the man is stupid and dull; it needs no slumber, when man must sleep or drop; ever at its post, ever ready for work, its alacrity never flags, its patience never gives in; its might is stronger than combined hundreds, and swifter than the flight of birds. . . .
—Samuel Butler
EREWHON



3719 Mantell Ave.
Cincinnati, Ohio 45236
(513) 891-4496

DOSFIX: A SET OF REPAIR JOBS FOR THE MODEL II (TRSDOS 2.0)

For the convenience of those users who wish to install some, but not all of our patches, we have separated the patches into individual files, named DOSFIX01, DOSFIX02, etc. In addition to the discussions in this document, each set of patches is preceeded in the DO files by PAUSE statements describing their function.

All of these updates, with the exception of #3, are applicable to 2.0 and 2.0A.

DOSFIX01: This one changes FORMAT so it will not try 10 times to format a track. FORMAT time is the time to find out if your diskette surface quality is marginal. We recommend that EVERYONE apply this patch, and if you get locked out tracks, you bulk erase the media, and try ONE MORE TIME ONLY. If a second attempt to format the diskette results in locked out tracks, we recommend that you send that particular diskette to our enemies behind the iron curtain as a 'CARELESS' package.

DOSFIX02: It seemed that Tandy's efforts to password protect their software just wasn't providing the protection they desired, so on 2.0, they modified the operating system such that NO password would work. Hmmm. This patch disables the new code, and the real password, KTA2, will now work.

DOSFIX03: These are from Fort Worth, and are offered without comment, except that they are not needed for 2.0A.

DOSFIX04: In 2.0, Tandy decreed that machine language programs could not load on top of any part of TRSDOS. I don't know whether they were trying to prevent a programmer from accidentally destroying his operating system, or simply trying to give software houses a tough way to go. In any event, the ability to overlay TRSDOS is AN ABSOLUTE REQUIREMENT FOR MOST SNAPP, INC. SOFTWARE. Put this one on for sure.

DOSFIX05: Undoubtedly for the same reasons mentioned in DOSFIX04, the TRSDOS DUMP command will not support addresses which overlay TRSDOS. This patch disables that address checking, but is NOT REQUIRED for any of our software. Unless you are interested in writing machine language programs, you probably shouldn't bother with this one.

DOSFIX06: Another module which checks addresses is DEBUG. If you would like to use DEBUG to fiddle with the 'forbidden' areas in the machine, put this one on. Not necessary for anyone.

DOSFIX07: It was a very poor design to put the BREAK key right next to the BACK SPACE key. Touch typist operators are continually reaching for the BACK SPACE key with their

pinky, and accidentally hitting the BREAK key. We can't repair the poorly designed hardware, but we can fix the software so another key does the job. With these patches installed, <ctrl>6 is the BREAK key, and the BREAK key does not work. Recommended for everyone.

DOSFIX08: I am sure that you were aware of the potential destruction of a diskette's directory on 1.2 if you changed a diskette at a bad time, and did not issue an "I" command. Tandy must have been very painfully aware also. I imagine the phone calls from this problem alone must have numbered in the thousands. Well, they FIXED the potential problem on 2.0, and the fix is called VERIFY DETECT (not to be confused with VERIFY). Unfortunatly, their fix slows disk I/O down so badly that many programs run TWICE AS LONG OR LONGER with VERIFY DETECT on. To make things worse, you can't even turn the facility completely off! These patches ALLOW you to turn VERIFY DETECT completely off, and make the NORMAL MODE of operation OFF. You will still be able to turn it on if you wish, but we think you probably will choose not to. Just be sure, as on 1.2, to issue "I" commands correctly.

DOSFIX09: The stepping rate, which determines how quicky the head will travel across the diskette surface, is selectable by the floppy disk controller to be 30ms, 20ms, 12ms, or 6ms. Tandy has chosen to use the 20ms rate. We have done a great deal of experimenting with this, and find that EVERY RADIO SHACK DRIVE works well at 12ms, which makes disk I/O much quicker. Please note that after you install these patches, that your drives will make a different sound than they did before. Some people say it sounds like a banshee. Don't worry about the noise. This update has been tested on about 700 Radio Shack drives, and has worked well on all but one. If it doesn't work on yours, you are probably out of luck, as the Shack won't have any sympathy for you.

DOSFIX10: This one simply eliminates the display of the Tandy Corp logo at boot-up time. If you like the pretty picture, don't put this one on.

DOSFIX11: This one will eliminate the TIME question when you boot up.

DOSFIX12: This one will eliminate both the DATE and the TIME questions when you boot up.

DOSFIX13: You may have noticed that while TRSDOS 2.0 has an 80 character type-ahead keyboard, that it doesn't work with BASIC. These patches correct that, but when they are installed, the HOLD key no longer functions with BASIC. We like it with the patches installed, and don't miss the HOLD key, but you can take your choice.

DOSFIX14: In our opinion, the FORMS command harasses the computer operator with too many B.S. questions. This one eliminates all the B.S.

DOSFIX15: If BASIC LPRINTs more than 255 consecutive characters without an explicit carriage return, BASIC sends one out at that point. This is clearly in conflict with the W= function of the TRSDOS printer driver, and produces confusing output. This one fixes BASIC to mind his own business.

DOSFIX16: You are probably aware that while TRSDOS supports 65535 logical records in a file, that BASIC supports only 32767. This is because BASIC converts the record number to an integer, which is correct, but then treats it as a SIGNED integer, which is NOT the way TRSDOS looks at it. This one allow you to access all 65535 logical records, but those numbered above 32767 must be expressed as a negative number. Just use this algorithm: GET (or PUT) buffernumber, recordnumber+(65536*(recordnumber>32767))

DOSFIXes: Courtesy of Snapp, Inc.

In our initial issue, we published a series of five patches which we had received without attribution. Subsequently, we discovered that these were part of a series which are copyrighted by Snapp, Inc. We now publish all seventeen of Bob Snapp's patches, preceded by a reproduction of his commentary on each patch. The patches themselves were reproduced directly from Snapp's distribution disk, so there is no opportunity for typographical error.

```
DOSFIX01:
   PAUSE FIX FORMAT SO HE IS NOT SO TOLERANT OF BROKEN MEDIA
   PAUSE THIS CHANGES HIS RETRY COUNT FROM 10 TO ONE
   PATCH FORMAT A=2942,F=0A,C=01
   PATCH FORMAT A=2973,F=0A,C=01
DOSFIX02:
   PAUSE ELIMINATE NON-PASSWORD MALARKEY FOR SYSTEM FILES
   PAUSE WITHOUT THIS PATCH, THERE IS "NO" PASSWORD FOR
   PAUSE TRSDOS SYSTEM FILES.  AFTER INSTALLING THIS PATCH,
   PAUSE THE MAGIC PASSWORD IS KTA2.
   PATCH SYSRES/SYS A=1786,F=23,C=00
   RESET
DOSFIX03:
   PAUSE NEXT THREE FROM FORT WORTH.  ELIMINATE FALSE ERROR 7
   PATCH SYSRES/SYS A=0E9C,F=7EB7,C=CB7E
   PATCH SYSRES/SYS A=0D45,F=91,C=99
   PATCH SYSRES/SYS A=0D94,F=91,C=99
   RESET
DOSFIX04:
   PAUSE NEXT TWO DISABLE LOAD AND TRANSFER ADDRESS CHECKING
   PAUSE THESE PATCHES ARE REQUIRED!!!! FOR MOST SNAPP SOFTWARE
   PATCH SYSRES/SYS A=15A7,F=30DF,C=0000
   PATCH SYSTEM/SYS R=2,B=198,F=3010,C=0000
   RESET
DOSFIX05:
   PAUSE NEXT SIX DISABLE ADDRESS CHECKING IN DUMP COMMAND
   PAUSE THESE PATCHES PRIMARILY OF INTEREST FOR MACHINE
   PAUSE LANGUAGE PROGRAMMERS.  THESE ARE NOT REQUIRED FOR
   PAUSE ANY OF OUR SOFTWARE, BUT WONT HURT ANYBODY.
   PATCH SYSTEM/SYS R=92,B=182,F=30EB,C=0000
   PATCH SYSTEM/SYS R=92,B=201,F=30D8,C=0000
   PATCH SYSTEM/SYS R=92,B=220,F=30C5,C=0000
   PATCH SYSTEM/SYS R=92,B=239,F=30B2,C=0000
   PATCH SYSTEM/SYS R=93,B=082,F=38D2,C=0000
   PATCH SYSTEM/SYS R=93,B=111,F=38B5,C=0000
   RESET
DOSFIX06:
   PAUSE NEXT SIX DISABLE ADDRESS CHECKING IN DEBUG
   PAUSE THESE ARE NOT REQUIRED FOR ANY OF OUR SOFTWARE
   PAUSE THESE ARE PRIMARILY OF INTEREST FOR MACHINE LANGUAGE PROGRAMMERS
   PAUSE BUT THEY WONT HURT ANYBODY
   PATCH SYSTEM64 R=2,B=80,F=38F3,C=0000
   PATCH SYSTEM64 R=2,B=88,F=30EB,C=0000
   PATCH SYSTEM64 R=6,B=183,F=30CE,C=0000
   PATCH SYSTEM64 R=6,B=191,F=38C6,C=0000
   PATCH SYSTEM64 R=7,B=211,F=38,C=18
   PATCH SYSTEM64 R=7,B=225,F=30,C=18
```

```
DOSFIX07:
   PAUSE NEXT SIX MOVE BREAK KEY FUNCTION TO <CTRL>6
   PAUSE THESE ARE REQUIRED FOR OUR BASIC MAPPING SUPPORT PACKAGE
   PAUSE THESE ARE RECOMMENDED FOR EVERYONE
   PAUSE EXCEPT SCRIPSIT DISKS.  DONT DONT PUT THESE ON A SCRIPSIT DISK!
   PATCH SYSRES/SYS A=0407,F=03,C=7E
   PATCH BASIC A=57E7,F=03,C=7E
   PATCH BASIC A=594C,F=03,C=7E
   PATCH BASIC A=5F70,F=03,C=7E
   PATCH BASIC A=6066,F=03,C=7E
   PATCH BASIC A=6515,F=03,C=7E
   RESET
DOSFIX08:
   PAUSE DONT LET THE WORD 'VERIFY' SCARE YOU.
   PAUSE WE ARE NOT! GOING TO TURN OFF WRITE VERIFY.
   PAUSE YOU ARE CORRECT, YOU WOULD BE A DAMNED FOOL TO TURN VERIFY OFF
   PAUSE BUT VERIFY DETECT IS A HORSE OF A DIFFERENT COLOR,
   PAUSE AND WE THINK MOST PEOPLE SHOULD TURN IT OFF COMPLETELY.
   PAUSE PLEASE READ TRSDOS 2.0 SUPPLEMENT FOR EXPLANATION OF
   PAUSE VERIFY DETECT.
   PAUSE WHEN VERIFY DETECT IS OFF, DONT VERIFY DETECT ON WRITES, EITHER
   PATCH SYSRES/SYS A=0D27,F=0A,C=03
   PAUSE BRING SYSTEM UP WITH VERIFY DETECT OFF
   PATCH SYSRES/SYS A=0040,F=20,C=22
   RESET
DOSFIX09:
   PAUSE CHANGE STEPPING RATE ON DISK DRIVES FROM 20ms TO 12ms
   PAUSE * CAUTION * THIS MIGHT NOT WORK WELL ON NON-RADIO SHACK DRIVES
   PAUSE BUT IT HAS BEEN TESTED ON ABOUT 1000 RADIO SHACK DRIVES
   PAUSE AND WORKED WELL ON ALL BUT TWO
   PAUSE AFTER THIS PATCH IS INSTALLED, YOUR DRIVES WILL MAKE A
   PAUSE DIFFERENT NOISE, WHICH HAS BEEN EQUATED TO A BANSHEE.
   PAUSE THIS IS TO BE EXPECTED.
   PATCH SYSRES/SYS A=0CFB,F=1E,C=1D
   PATCH SYSRES/SYS A=0D12,F=1E,C=1D
   RESET
DOSFIX10:
   PAUSE KISS THE TANDY LOGO GOOD-BYE
   PATCH SYSRES/SYS A=24D3,F=00,C=3B
DOSFIX11:
   PAUSE ELIMINATE THE TIME QUESTION AT BOOT-UP (OPTIONAL)
   PAUSE IF YOU DONT WANT TO ELIMINATE THIS, BREAK OUT NOW
   PATCH SYSRES/SYS A=2390,F=0A,C=39
DOSFIX12:
   PAUSE ELIMINATE THE DATE QUESTION AT BOOT-UP (OPTIONAL)
   PAUSE IF YOU DONT WANT TO ELIMINATE THIS, BREAK OUT NOW
   PATCH SYSRES/SYS A=2367,F=73,C=CA
DOSFIX13:
   PAUSE UPDATE BASIC TO SUPPORT 2.0 KEYBOARD BUFFERING
   PAUSE DOSFIX07 IS A PREREQUISITE TO USING THIS FIX
   PAUSE IF YOU ELECTED NOT TO INSTALL DOSFIX07, DO NOT ATTEMPT
   PAUSE TO INSTALL DOSFIX13
   PAUSE THIS FIX MAY BE USED ALONE IF YOU MODIFY IT SUCH THAT BOTH
   PAUSE OCCURENCES OF THE HEX CODE '3E7E' ARE RI LACED BY '3E03'
   PAUSE (THERE ARE TWO)
   PATCH BASIC A=325C,F=49,C=62
   PATCH BASIC A=57DC,F=49,C=62
   PATCH BASIC A=60EF,F=49,C=55
   PATCH BASIC A=6049,F=00000000000000C53E04CF,C=3A6161B728063E7EF5AF18
   PATCH BASIC A=6054,F=2005F6FF78C1C9C13A6161B7,C=17C53E04CF20043C78C1C9AF
```

```
   PATCH BASIC A=6060,F=C8AF3261613E7EB7,C=C1C93A6161B720E7
DOSFIX14:
   PAUSE ELIMINATE B.S. QUESTIONS FROM FORMS COMMAND EXECUTION
   PATCH SYSTEM/SYS R=42,B=172,F=CD07,C=1809
   PATCH SYSTEM/SYS R=42,B=225,F=CF20FB,C=000000
   PATCH SYSTEM/SYS R=42,B=231,F=CD0725,C=C3FB22
DOSFIX15:
   PAUSE ELIMINATE GARBAGE CARRIAGE RETURN TO PRINTER FROM BASIC
   PATCH BASIC A=56F8,F=CC0D572807,C=0000000000
DOSFIX16
   PAUSE UPDATE BASIC TO SUPPORT 65535 LOGICAL RECORDS PER FILE
   PATCH BASIC A=660C,F=0933,C=DC3C
DOSFIX17:
   PAUSE DOSFIX17: Correct error in BASIC relating to hex constants.
   PATCH BASIC A=39DD,F=237E,C=EF00
```

Printed on an NEC 7725 Spinwriter with Courier 10 thimble.

GREAT PRICES . . . GREAT PRODUCTS

TRICOR 1-800-253-4358

Quality Scotch® Diskettes

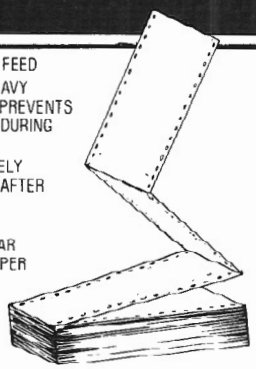
The recording medium has a special formulation of oxides and polymers specifically developed to withstand repeated wear. The low abrasion factor (each diskette is twice as smooth as the industry standard) built into each Scotch Diskette provides greatly extended life for the read/write heads.

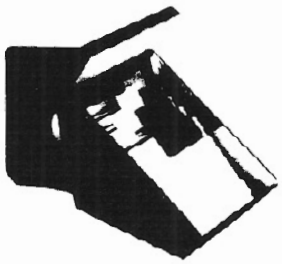
CATALOG NUMBER	DISK SIZE	FORMAT DESCRIPTION	HARD SECTORS	QTY/ BOX	LIST PRICE	TRICOR PRICE	10 BX OR MORE
DK-741-0	8"	Sgl S/Dbl D	Soft	10	$61.00	$35.00	$34.00

Self-Adhesive Address Labels

Choice of two sizes in quantities matched to your needs . . . from as few as 1,000 labels to 20,000 labels per box, and in formats from 1 across to 4 across. The backing is extra strong to prevent skewing in the printer and to allow for accurate refolding.

CATALOG NUMBER	LABEL SIZE	NUMBER ACROSS	CARRIER WIDTH	LABELS/ BOX	LIST PRICE	TRICOR PRICE
Self-Adhesive Address						
LA-4139	3½ x 15/16	1	4.25"	1000	$ 8.60	$ 7.25
LA-4013		1	4.25"	5000	$20.85	$14.50
LA-4030		2	8.1"	10000	$41.50	$31.28
LA-4031		3	11.75"	15000	$62.25	$42.97
LA-4029		4	15.3"	20000	$83.00	$57.30
LA-4141	4 x 1 13/16	1	4.75"	1000	$13.25	$ 9.85
LA-4142		1	4.75"	3000	$34.82	$25.65
Piggyback						
LA-4097	3½ x 15/16	1	4.25"	3000	$81.96	$53.16
Large Stock Shipping						
LA-4076	5 x 2 15/16	1	5.75"	3000	$75.93	$52.85





Flip-N-File Stores 50 Disks

Attractive, durable, dust-free protection for both 5¼" and 8" disks. The Flip Sort is one of the most popular storage containers. Formed from a highly durable, transparent smoke colored acrylic, its elegant design enhances any office.

Index dividers are included to provide easy retrieval and can be set to any desired position.

The unique hinged lid provides easy access and doubles as a carrying handle.

CATALOG NUMBER	DESCRIPTION	CAPACITY	LIST PRICE	TRICOR PRICE
ST-5FS	For use with 5¼" Disks	50-60	$31.00	$25.85
ST-8FS	For use with 8" Disks	50-60	$45.00	$35.85

Plain White Word Processing Paper

(Heavy 20# bond)

Plain white, 20# bond, printout paper for unlined reports and all word processing uses. We offer this unlined paper in 9½" x 11" sizes . . . with perforated tractor feed margins. Remove the perforated margins, you have a standard 8½" x 11" sheet . . . great for word processing, ideal for any business use.

CATALOG NUMBER	SIZE W x L	PARTS/ SET	SETS/ BOX	LIST PRICE	TRICOR PRICE	BULK BOX 3000 SHTS
PA-0C591	9½ x 11	1	1000	$23.80	$15.85	$35.50

SPECIAL INSTRUCTIONS	QUANTITY	ITEM NO	PRODUCT DESCRIPTION	AMOUNT
	1			
	2			
	3			
	4			
	5			

1-800-253-4358
in Michigan
616-375-7519

VISA

SUBTOTAL	
TAX Mich res add 4%	
SHIPPING	
TOTAL	

Call Toll-Free Today!

PAYMENT BY

- ☐ VISA
- ☐ Master Charge
- ☐ C.O.D.
- ☐ Check Enclosed

card number

signature

Name

Company

Address

City State Zip

Reader Survey Volume 1, Number 2 July-August, 1982

Please return to **two/sixteen,** Box 1216, Lancaster, PA 17603.

This survey seeks your opinion about several matters:

1. Where would you prefer to have a computer dial-up information exchange based?

____CompuServe ____The Source ____II/16 (Lancaster, PA dial-up)
____doesn't matter ____not interested in exchange

2. We are considering establishing a software retail facility which would market only to users of Model IIs and 16s. It would sell at competitive discounts. We don't know whether or not it could carry Tandy software, but it can obtain most other products. The primary reason we are considering this is that we believe we can provide a better after-sale service than Model II or 16 owners can expect to get from vendors who sell software for dozens of computer models. What do you think about such a service?

____I like the idea
____I don't like it
____I don't care

If such a facility were started, which software would you like to see carried?

____Radio Shack software
____Pickles & Trout CP/M
____MicroPro products: ____WordStar ____SpellStar ____Mailmerg
____SuperSort ____DataStar ____Wordmaster
____Utilities, such as SNAPP, Racet, etc., for TRSDOS
____Utilities for CP/M
____Data Base managers
____Others (be specific as possible): _____

3. Should we publish consumer information, such as where you can buy a specific product at the lowest price (e.g.: Consumer tip for July: Wabash 8" double density diskettes from ABM Products, 8868 Clairemont Mesa Blvd., San Diego, CA 92123, telephone (800) 854-1555 [in Calif.: (714) 268-3537]. We bought 100 double-sided at $3.09 each and 50 single-sided at $2.74. UPS shipping costs will add about 2–4%. Best local price is well over *twice* this amount.)?

____should carry such information
____don't bother

Optional responses (fill in all, any part, or none):

Name: _____

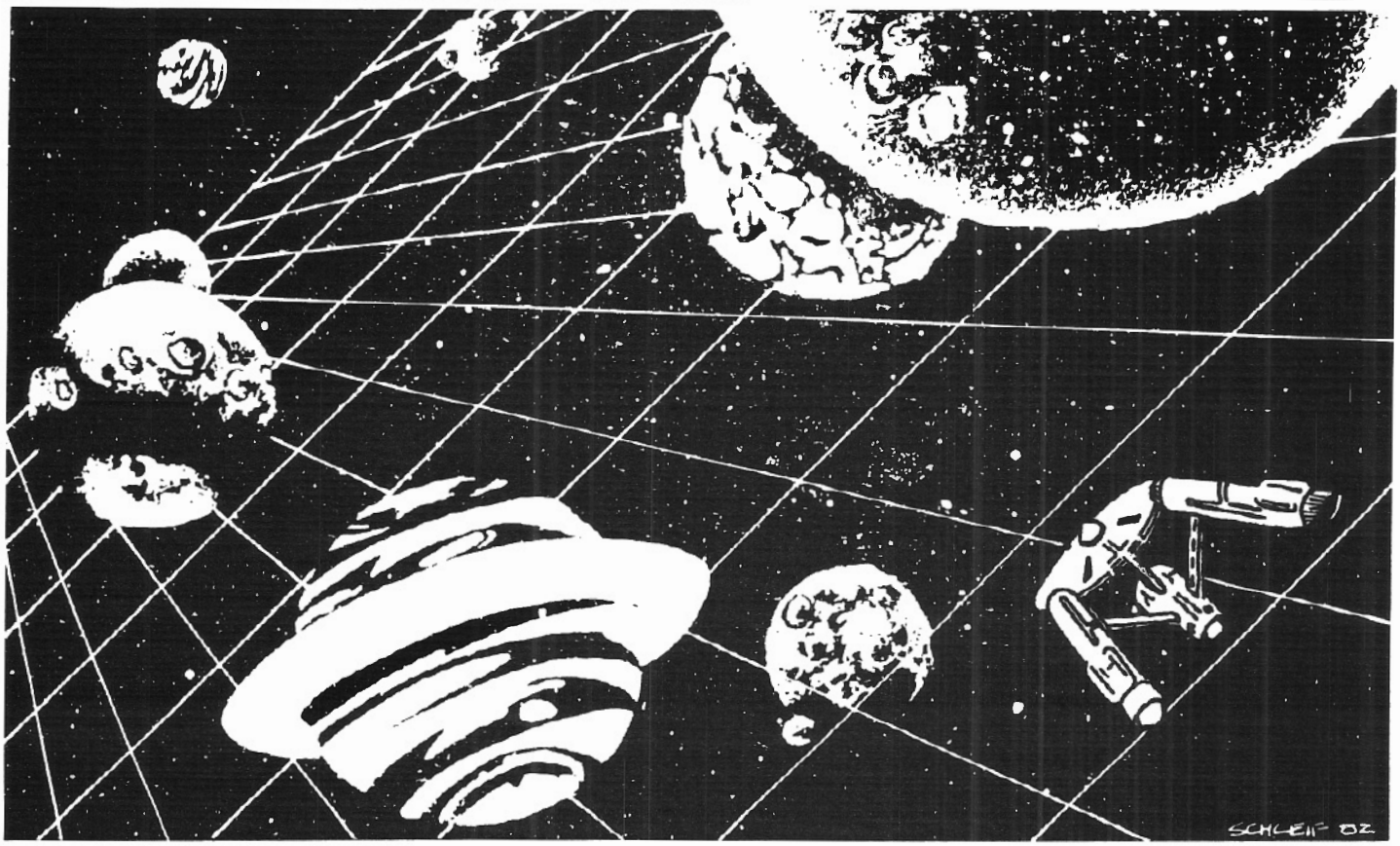
Address: _____

City, State, ZIP _____

Occupation: _____

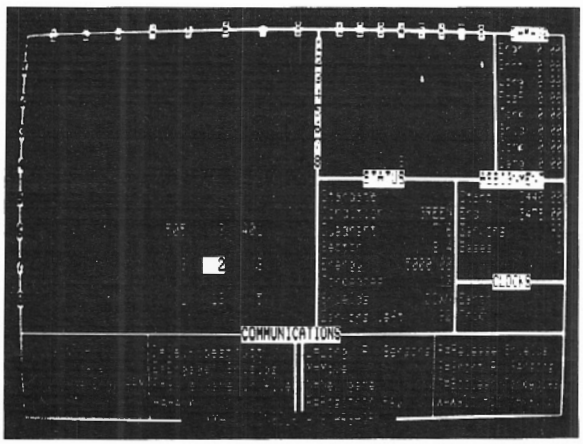
Some vendors have expressed interest in sending direct mailing offers to our subscribers. Check here if you wish to be excluded from such mailings: ____

STARSHIP I



STARSHIP/I (copyright registered) is a save-the-galaxy type game based on the venerable STARTREK games which have been played on large mainframes for more than ten years. It is specifically designed for the RS Model II and features a constant console display with no scrolling. Each game begins in a randomly arranged galaxy of 64 "quadrants," each containing 64 "sectors."

STARSHIP/I is a "strategic" game (as opposed to "tactical" games which involve reflex action but little thought). However, STARSHIP requires *quick* thinking, as all events are timed. Because STARSHIP is written in the RS BASIC interpreter language, users can modify the source code or use sections of the code as models to learn techniques which can be used to produce games of their own design.



starship I

STARSHIP comes with a disk instruction file which can be listed on your printer. It can also be listed on the screen at the beginning of a session. In addition, STARSHIP contains a few undocumented "surprises" at advanced levels of play (there are ten levels of difficulty). These surprises may be analyzed by reviewing the source code, then modified or deactivated if desired.

Games may be saved in files at any point for later play or for "duplicate" games in which several players start with the same layout and position and compete for the highest ending score.

STARSHIP/I was sold for several months at $50 through leading computer magazines. It is now available to **two/sixteen** subscribers at the special price of $30, including air mail (anywhere) in protective Kangaroo mailer.

Let CP/M® improve your TRS-80™

Increase the speed, power and memory of your TRS-80 Mod II and Mod 16 with JobStream™ CP/M 2.2 from Aton International.

It's easy to upgrade to CP/M—just plug in the JobStream diskette. JobStream is two to five times faster than the competition, while disk reliability is improved. And the extra memory space JobStream provides for your programs increases your speed even more. With BASIC Surrogate and VisiCalc™ Surrogate interfaces you can use your TRSDOS™ BASIC and VisiCalc programs in the JobStream CP/M environment. And the Mod 16 is supported—including the double-sided disk drives.

JobStream is currently available in two levels. Level I requires 64K RAM and Level II requires 96K RAM* Level II allows programs to access other 32K memory banks, makes more memory space available to user programs and adds disk track buffering to Level I support.

So send in the coupon below today and get much more out of your TRS-80 Mod II and Mod 16.

*Radio Shack dropped the price of its 64K add on memory board for the Mod II from $715 to $399 and made it a standard catalog item (#26-4105). It can be installed at any Radio Shack Service Center for $30.

Aton

ATON INTERNATIONAL, INC. 260 Brooklyn Avenue, San Jose, CA 95128 U.S.A. (408) 554-9922

Yes! I want to improve the performance of my Mod II or Mod 16 with Aton products. Please send me:

- ☐ JobStream CP/M Level I (requires 64K RAM). $179
- ☐ JobStream CP/M Level II (requires 96K RAM). $235
- ☐ BASIC Surrogate. $99*
- ☐ VisiCalc Surrogate. $99*
- ☐ OmniWriter™ Video Text Editor. . . . . $150* (designed specifically for the JobStream CP/M environment)

*SPECIAL OFFER! Save $25 on each software package if ordered now with any level JobStream CP/M.

Total Price $_____

Name_____

Address_____

_____

Phone_____

☐ Check enclosed ☐ VISA ☐ M/C

Card #_____ Exp. Date_____

Signature_____

California residents add sales tax. Shipping and handling charges will be added to VISA and MasterCard orders.