Basic Computing

The TRS-80 User Journal $3 per copy July, 1983

FORMERLY 80 U.S.

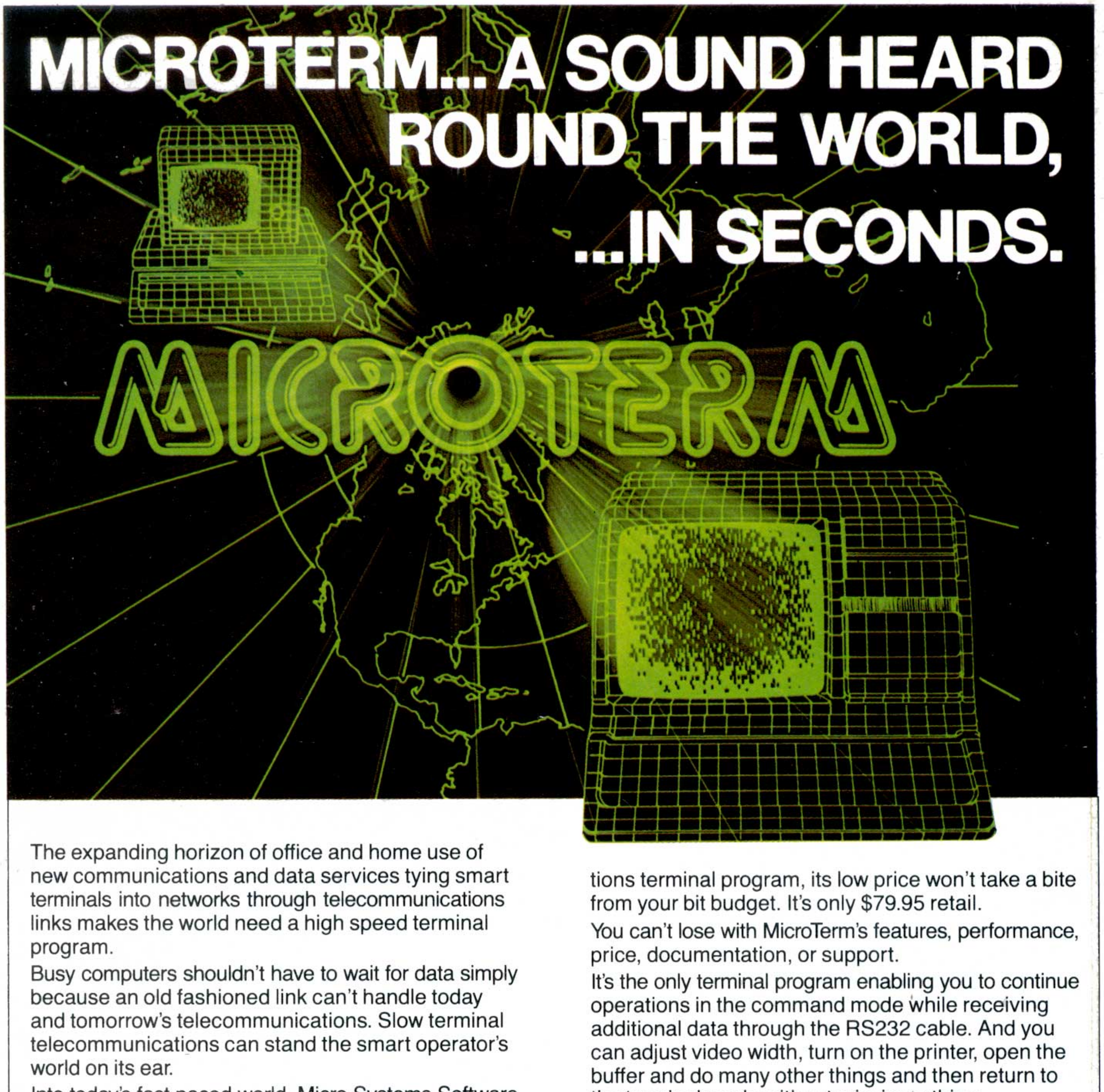
MAKING MUSIC WITH YOUR COMPUTER

COLOR COMPUTER HARMONY
BY LEO CHRISTOPHERSON

RECOVERING LOST
MODEL II PROGRAMS

0 60959 07
0 318585

MICROTERM... A SOUND HEARD ROUND THE WORLD, ...IN SECONDS.



The expanding horizon of office and home use of new communications and data services tying smart terminals into networks through telecommunications links makes the world need a high speed terminal program.

Busy computers shouldn't have to wait for data simply because an old fashioned link can't handle today and tomorrow's telecommunications. Slow terminal telecommunications can stand the smart operator's world on its ear.

Into today's fast paced world, Micro Systems Software presents MicroTerm, the first truly high speed terminal for this inquisitive world. Some computers can run at up to 4800 baud null-free (9600 in some direct-connect applications).

MicroTerm enables you to do more in less time, in both ASCII and the new "error-free" direct file mode. Its unique Macro-Key function allows you to have 10 user-defined keys that transmit up to 64 characters at a single stroke. You can even dial a phone number and transmit the buffer at a specified time completely unattended by the operator.

And while MicroTerm improves your computer's "ears" by outperforming any other telecommunications terminal program, its low price won't take a bite from your bit budget. It's only $79.95 retail.

You can't lose with MicroTerm's features, performance, price, documentation, or support.

It's the only terminal program enabling you to continue operations in the command mode while receiving additional data through the RS232 cable. And you can adjust video width, turn on the printer, open the buffer and do many other things and then return to the terminal mode without missing a thing.

Available for the TRS-80 Models I, II, III, 16, IBM PC, Zenith Z-100, and Apple II computers.

If these advantages are what you want in your world, communicate with your nearest MicroTerm dealer. For information contact: Micro Systems Software, Inc., 4301-18 Oak Circle, Boca Raton, Florida 33431,
Telephone Toll Free: 1-800-327-8724
In Florida (305) 983-3390

MICRO-SYSTEMS SOFTWARE, INC.

4301-18 Oak Circle, Boca Raton, Florida 33431, Telephone: (305)983-3390
Toll Free 1-800-327-8724

"THE RESULTS ARE IMPRESSIVE..."
—Dennis Kitsz, 80 Microcomputing; 12/82

Langley-St. Clair's* *Soft·View*™ Replacement CRT's

eliminates the strobe, flicker and fatigue from TRS-80's.™

Now you can upgrade your monitor with the new medium persistence green or amber phosphor tube.

State-of-the-art systems such as IBM™ and Apple III™ do not use the less costly "P4" B&W display tube because it is actually intended for TV viewing and its rapid strobes (60 times per second) cause irritating eye fatigue.

No amount of "green plastic" will solve this problem. But the new ***Soft·View*** CRT display tube from Langley-St. Clair will.

- Available in slow decay Green or medium decay "European Amber" (the standard in Europe)
- Made with Lead/Strontium impregnated glass that stops X-ray emission.
- Of high-contrast face glass that also stops most U.V. radiation.
- Available in frosted glass with extra Anti-Glare benefits.
- Easily installed...comes with pre-mounted hardware.
- Warranted for one full year against manufacturing defects or tube failure.
- The finest quality double-dark glass phosphor fields to produce dramatic contrast.
- Ideal for Word Processing and Programming, yet fast enough for Games and Graphics.

LSIS *Soft·View*™ CRT'S

☐ #GN42 Green Phosphor	$79.95
☐ #GN42G Green Phosphor w/Anti-Glare	$89.95
☐ #OR34 Amber Phosphor	$89.95
☐ #OR34G Amber Phosphor w/Anti-Glare	$99.95
also available:	
☐ #R22G Red Phosphor w/Anti-Glare	$139.95
☐ #B22G Blue Phosphor w/Anti-Glare	$139.95

Plus: $7.00 for packing and UPS Shipping
$17.00 for Overseas, Parcel Post or UPS Blue Label
Add Sales Tax where applicable.
(Inquire about the CRT's we have available for many other computer models)

For MasterCard and Visa Orders only, call
800/221-7070 (in N.Y. call 212/989-6876)

Langley-St. Clair Instrumentation Systems, Inc.
132 West 24th St., New York, N.Y. 10011

*World's largest supplier of upgraded replacement CRT's.

 Please address correspondence to: **Basic Computing, 3838 South Warner Street, Tacoma, Washington 98409, telephone (206) 475-2219.**

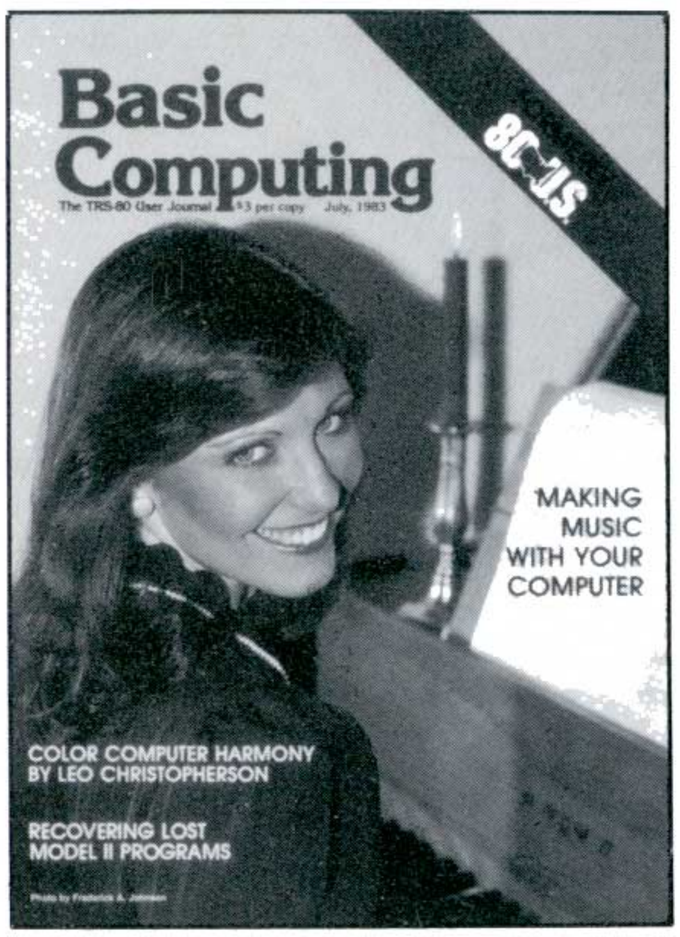
Advertisers: Basic Computing will accept relevant commercial advertising which pertains to, or is for use on, the Tandy Corporation microcomputers. Write for a current rate schedule.

Authors: We constantly seek material from contributors. Send your material (double spaced, upper/lower case, please) and allow approximately 4 to 6 weeks for review. Programs must be supplied in machine readable form on diskette or tape, clearly marked as to model and operating system. Text files may be on diskette. Media will be returned if return postage is provided. Cartoons and photographs are welcome. Generous compensation will be made for non-trivial works which are accepted for publication. Basic Computing pays upon acceptance rather than on publication.

SUBSCRIPTION PRICE: U.S.: $19.97 for one year, $34.97 for two years and $49.97 for three years. **Canada and Mexico:** $29 per year, no two or three year subscriptions are offered. **All other:** $36 per year via surface mail, $78 per year via airmail. Two and three year subscriptions are not offered.

ISSN Publication #0199-1035. Basic Computing is published monthly by 80-Northwest Publishing, Inc., 3838 S. Warner St., Tacoma WA 98409-4698. Printed in the United States of America.

POSTMASTER: Please send change of address form 3579 to Basic Computing, 5615 West Cermak Road, Cicero, Illinois 60650. **Second Class postage PAID at Tacoma, WA and additional entry points.**



This month's model is Claudette Hensley of Tacoma, Washington. Our photographer was Frederick A. Johnsen, also from Tacoma.

Basic Computing

The TRS-80 User Journal

TRS-80 is a trademark of the Tandy Corp.

Vol. VI, No. 7 — July, 1983

ARTICLES

A Star is born...

The PRODUCER™

The Professional Program Writer.

Now Available for the Model I & Model III
With 48K and 2 Disk Drives
(Please Specify Model I or III)

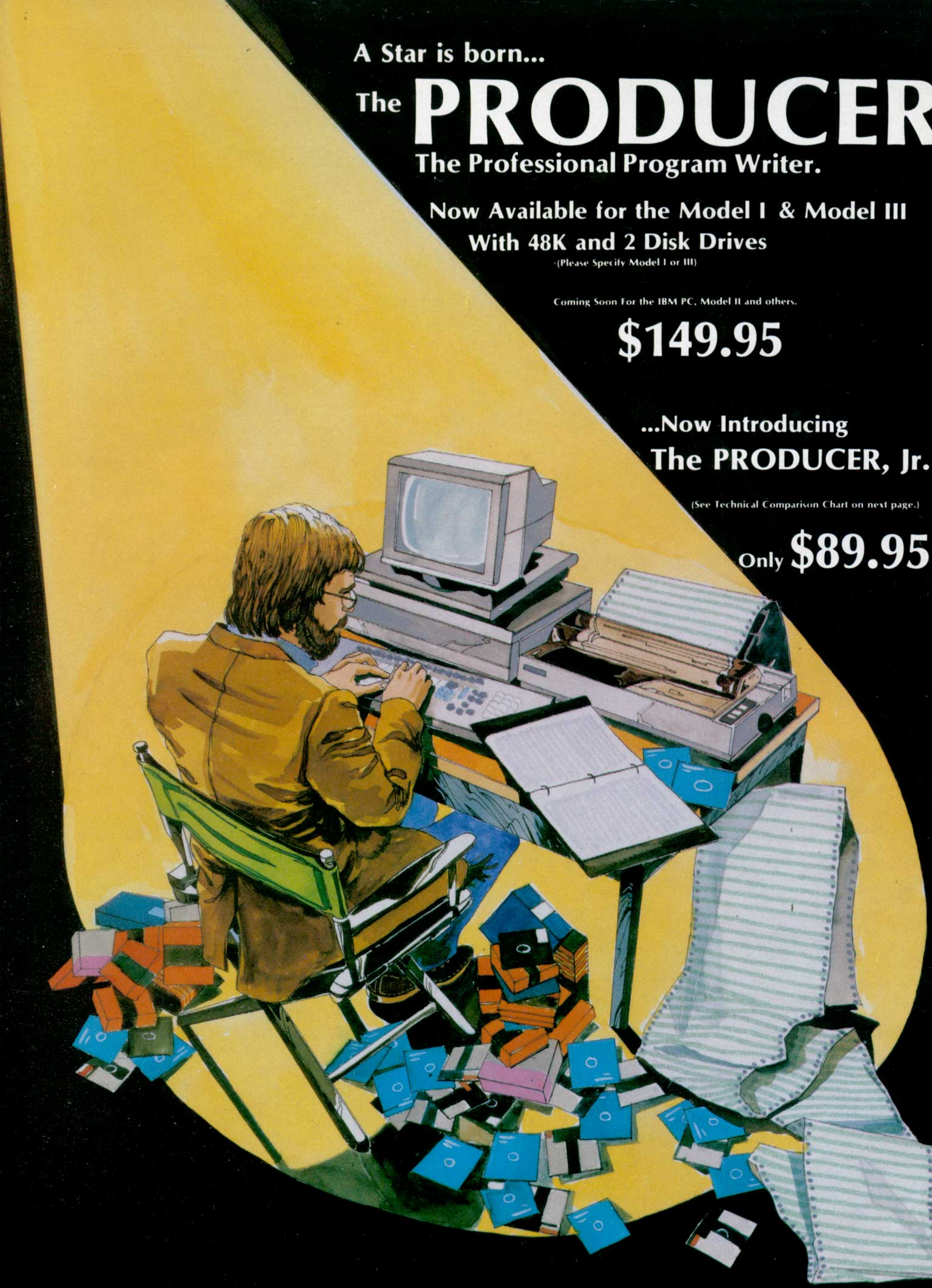
Coming Soon For the IBM PC, Model II and others.

$149.95

...Now Introducing
The PRODUCER, Jr.

(See Technical Comparison Chart on next page.)

Only $89.95



If BASIC somehow doesn't seem so basic, and your software has been getting you nowhere, then let me introduce myself. THE PRODUCER is my name. Writing programs is my game. If you're interested in an inexpensive way to quickly and easily write professional quality custom software programs, then read on.

Have you ever wasted money on software that didn't do what you thought it would? Are you burned out on high priced, canned programs that make big claims, but don't meet your needs. Has a lack of good software made your computer an expensive gadget that doesn't do what you hoped it would? If so, I'm here with good news. With my help you can put those problems behind you.

What is THE PRODUCER?

In short, I'm a Program Generator designed to write customized software programs. If you have a need to store and retrieve information, perform calculations on your data and get displayed or printed reports, I can help you develop a program to do just that, in just about any format you can imagine. That's why I'm called a program development system. I'm a powerful and sophisticated software package, born of vast technical knowledge and professional design experience. Yet, I'm the most simple, practical, easy to use and functionally versatile program generator ever put on the market.

What can THE PRODUCER do for you?

How can I help you? Just let me count the ways.

1. I make programming easy. I'm user friendly. If you can answer simple English questions and push buttons, I'll do all the hard work. Let me worry about the BASIC language required to complete your program. Describe the program you want and I'll write it for you. The process is simple. First, I produce a printed planning sheet to help you get your ideas on paper. When you've finished planning, you're ready to draw your screen right on the monitor. The Screen design and appearance is completely at your command, including graphics. You have full editorial control. Make all the changes you like, until you're satisfied. After you're thru, I'll guide you thru some simple English questions about your Screen. Based on your answers, I'll write a complete BASIC program for you.

2. I can save you 100's of hours of time. I provide real short cuts to meet your needs by going direct from your idea to a ready to use, customized program. I write all the BASIC code for you. I'm simple, but I'm not shallow. The only limit to my capacity is your imagination.

3. I can save you big bucks. You may never need to buy another program to store and retrieve your data. With my help, you can design your own custom programs to get your job done.

4. I can help make you more productive. Having me as your partner will be like taking a smart pill. I'll pay for myself many times over by increasing your productivity and creativity. With my custom programs serving you, your computer will be the valuable friend you hoped it would be.

5. I can help make you a real pro. Based on your ideas, I write complete stand alone programs. I write in BASIC code, but you talk in English to me. And if you're an entrepreneur, you can sell the programs you and I create. As a licensed owner of THE PRODUCER you may do so without paying royalties.

6. I can make good programmers much better. I produce fully commented BASIC code so you can use me as a building block to write your own specialized software. I have the best screen and input module available anywhere at any price.

What do you get with THE PRODUCER?

1. Two diskettes containing THE PRODUCER program development system, and a complete disk operating system. Also included is a free sample personal management program (value $59.95) useful to every computer owner.

2. Extensive documentation in a three ring binder, with index tabs, quick reference system, comprehensive index and sections for the novice and professional programmer.

3. Complete printed tutorial, walking through each step in THE PRODUCER process and resulting in a finalized sample program. An audio cassette tape ($14.95 value) of the tutorial session is also provided free for a limited time.

4. A toll free technical assistance number for PRODUCER owners.

5. A free one year subscription to THE PRODUCER's quarterly newsletter containing ideas, sample programs and update information related to THE PRODUCER.

TECHNICAL COMPARISON CHART

	PRODUCER	JR.	QUICKPRO	CREATOR
FEATURES OF THE FINISHED BASIC PROGRAM				
Full Screen Oriented Input of All Fields	YES	YES	NO	NO
Edit without Retyping with insert & delete	YES	YES	NO	NO
Restrict field Length automatically	YES	YES	YES	NO
Unlimited Restriction choice for each field	YES	YES	NO	NO
User defined Custom Prompts for each Field	YES	YES	NO	NO
Full Speed Typing in ALL Fields	YES	YES	NO	YES
Immediate Exit from any Field to Menu	YES	YES	NO	YES
Enter Fields from last Record automatically	YES	YES	NO	NO
Fast BTREE File Structure (No Sort Needed)	YES	YES	NO	NO
Find Record with Part of a Key	YES	YES	NO	NO
Hi-Speed Global Search for ANY Field in a Record	YES	YES	NO	NO
Duplicate Keys and Multiple Keys Supported	YES	YES	limited	NO
Global Field Replacement Function	YES	YES	NO	NO
Run Predefined Reports from Finished Program	YES	YES	NO	NO
Select Reports from Menu in Finished Program	YES	NO	NO	NO
Sort (machine language) ANY Field-Free	YES	NO	NO	NO
Custom Mailing Labels Option (any Size)	YES	NO	NO	NO
Do Calculations on fields in Program	YES	YES	YES	NO
Sell Finished Program with No Royalty	YES	YES	NO	NO
PRODUCER CAPABILITIES & FEATURES				
Toll Free Question Line	YES	YES	NO	NO
Create PROFESSIONAL Finished Program	YES	YES	NO	NO
Modify Program without Starting Over	YES	YES	NO	NO
Ease of Use, including Complete TUTORIAL	YES	NO	NO	NO
Number of Calculations allowed per field	8	8	1	0
Use Field Names for Calculations	YES	NO	NO	NO
Use ALL Math Functions in Calculations	YES	NO	NO	NO
Generates a BASIC Program	YES	YES	YES	YES
Custom Design exact Screen YOU desire	YES	YES	NO	NO
Full Feature Screen Generator (graphics)	YES	NO	NO	NO
Easy Report Generation with Any Restrictions	YES	NO	NO	NO
Complete & Thorough DOCUMENTATION	YES	YES	limited	NO
Detailed Quick Reference Materials	YES	YES	NO	NO
Audio Cassette Tutorial Available	YES	NO	NO	NO
Program Planning Form Provided	YES	NO	NO	NO
Sample Programs Available before Purchase	YES	YES	NO	NO
FREE UTILITIES INCLUDED				
Free Menu Driven DOS Utility Package	YES	YES	NO	NO
Free Disk Operating System (Super Fast)	YES	YES	NO	NO
File Rebuilder & Reorganizer included Free	YES	NO	NO	NO

Send Cashier's Check, Money Order or Certified Check to: **$149.95**

PRODUCER SOFTWARE
A division of Texas Computer Systems, Inc

P.O. Box 1327 Arlington, Texas 76004-1327
Or For Master Card - Visa - American Express
Call 1-800-433-5184 Texans Call 817-274-5625

PRODUCER Generated Programs

We understand your reluctance to invest in THE PRODUCER until you know for sure it provides what we say it does. The programs below are unmodified, finished programs generated by THE PRODUCER. Our customers tell us that PRODUCER generated programs are better than many high priced programs written by human professionals. Compare these programs to any you have seen selling at a similar price. Their quality will surprise you. Buy any of these PRODUCER generated programs now and we will gladly apply the cost of that program toward the future purchase of THE PRODUCER. Or buy THE PRODUCER now and select one of the PRODUCER generated programs to be included in your order, absolutely FREE.

Executive Scratch Pad	$29.95
Expense Notebook	$19.95
Teacher's Grade Book	$29.95
Recipe Program	$19.95
Photo Assistant	$29.95
Video Library	$29.95
Mailing List	$19.95
Personnel Program	$39.95

See descriptions in March and April Issues of 80 Micro

Inventory 1	$39.95
Maintains group of identical items	
Inventory 2	$39.95
Maintains record for individual items	
Record Library	$19.95
Organizes your music center	
Reader's Guide	$19.95
Orgainzes magazine articles/clippings	
Personal Checking	$29.95
Organizes checking accounts	
Loan Calculator	$39.95
Calculates loans, principal and interest	

The PRODUCER, JR.

If you don't need all the features of The PRODUCER-- the professional program writer and you don't want to spend $149.95 on software, READ ON. Producer Software has the perfect solution for you.

THE PRODUCER Jr.

A SIMPLE TO USE professional program generator. And it's only $89.95. This self documented program generator can have anyone creating self-contained stand alone programs in no time at all. No other program generator can provide you with ease of use, low price and quality.

FEATURES

- screen generator allows easy definition and creation of fields
- B-tree file lets you search, insert, delete and edit
- reports by single restriction (no sort)
- much more (see technical comparison chart)

The PRODUCER Jr. comes to you complete with all the above features and its own operating system for only $89.95. But don't let this low price fool you--The PRODUCER Jr. is still the most powerful program generator on the market today with the exception of the original PRODUCER. And The PRODUCER JR. is completely compatible to the original PRODUCER, so if you decide to upgrade at a later date you may do so for only $75.00. So don't miss out. Order today.

For an independent product review of the PRODUCER see page 62 of March issue of 80 Micro

 For more information write for our free brochure

Basic Computing
The TRS-80 User Journal

Publisher
I. Mike Schmidt

Managing Editor
Cameron C. Brown

Associate Editors
Terry R. Dettmann
Spencer Hall
Jim Klaproth

Contributing Editors
Timothy K. Bowman
James A. Conrad
Bob Liddil
Thomas Quindry

Advertising Coordinator
Mark Metzger

Advertising Representatives
East of the Mississippi river
Garland Associates:
John A. Garland
Frank Surace
(617) 934-6464/6546

Advertising Representatives
West of the Mississippi river
Jules E. Thompson, Inc.
Pacific Northwest (415) 348-8222
Mountain States (303) 595-9299
Pacific Southwest, Texas and Oklahoma (213) 378-8361

Promotion/Circulation
Robert P. Perez
Julie Bartz

Production
Catherine D. Doud

Editorial Secretary
Eva R. Jones

Accounting
Helen Dalton

MOVING?
Please enclose your label or write your name and address as it appears on your label.

Name _____

Address _____

City/State _____

Write in new address:

Name _____

Address _____

City/State _____

Basic Computing
Subscription Department
5615 West Cermak Road
Cicero, IL 60650

Directions

By I. Mike Schmidt, Publisher

It's been about six months since I last wrote about the "state of the business" and many things have happened since then. Back in January, we were all excited about going slick. That's a fact now, and after a short trip back to the printer in Minnesota, Cathy Doud has learned more about manipulating color pages and layout to our best advantage.

We recently acquired distribution through *Waldenbooks* and just before the end of last year, through *B. Dalton Booksellers*. It never fails to amaze me that after five years, there are still thousands of TRS-80 owners who have not heard of us. Actually, that is our problem, isn't it? Apparently, you have to wave your flag long and hard to get attention.

I have always been somewhat disturbed when people ask what the name of our publication is. When I tell them, they usually respond with: "What does that mean?"

Checking some of the local bookstores with magazine racks recently, I found *80-U.S.* was not with the other computer magazines but, rather, with the travel mags. It happened enough times to make it apparent that we simply did not have a definitive name.

Hanging on the wall in my office is a prototype cover of a magazine we were thinking about back in September, 1981. For various reasons, that idea never got off the ground, but the name on that cover was *Basic Computing*. The name was concocted by Tom Huber, Terry Dettmann and myself. It seemed like a natural, and we were amazed to find no one else using it. It also fits our editorial direction, since we have long been stressing tutorials which help users learn more about their computers. Cam Brown recently stated our editorial goals in three simple words: "Enlighten, educate and entertain."

That is how the new name, *Basic Computing*, came about. We hope you agree that it is a step in the right direction. Now, when someone asks what our name is, they can immediately comprehend and say: "Oh, a computer magazine".

About the time you see this in print, we will be launching a rather large direct-mail promotion. Please do not be upset if you are already a subscriber and get such a promotional piece in the mail. The labels for large lists are applied by machine and are in zip code order. We do not have access to the computer which creates the list so we can't purge it of our own subscribers. If you should get such a piece of mail, do us a favor and give it to a friend who has not heard of us.

We are also pleased to announce that starting with this issue, Jules E. Thompson, Inc. will be representing *Basic Computing* for advertising. They will cover all states west of the Mississippi River. You may recall that Jules E. Thompson was the advertising representative for *Creative Computing* magazine from its beginning until it was taken over by Ziff-Davis. Starting with the next issue, Jack Garland Associates will represent us in all states east of the Mississippi River. That about covers the country, doesn't it? We welcome these two fine firms and hope their association with us will be enjoyable.

As a reader, you may well ask what that will do for you. Simply stated, it means more advertising, more pages of editorial material and a better magazine all around.

Basic Computing

SUBSCRIPTIONS

I've never heard of Basic Computing. Why?

You've been missing something. Since 1978, we were called **80-U.S. Journal**. We have now changed our name to more accurately describe what we write about. We are computerists who publish a journal, not publishers talking about computers.

What is Basic Computing?

We are a monthly magazine covering all models and aspects of the TRS-80 microcomputers. Each issue contains a mix of articles and programs for every level of expertise in the computing field.

We have regular columns and departments to help both the beginning Color Computerist and the advanced Model III assembly language programmer. We make a special effort to make our publication understandable to beginners and advanced computerists alike.

What makes Basic Computing special?

We give complete program listings that are from working programs, not just bits and pieces of computer code. Material in our journal comes from actual computer users, not writers who have little hands-on experience with your model. We discuss and give working programs for every model of TRS-80. If you own a TRS-80 Model I with exotic hardware additions, or use a Model 100 to communicate to a Model 16B, we have information you need.

What does it cost?

Only $19.97 for a full year's subscription. That's less than $1.67 per month for one of the most informative magazines you can buy for you and your computer. Even greater savings are possible with longer subscriptions.

Ok! I want to subscribe. What do I do?

It's easy. Fill out the attached order card and drop it in the mailbox.

BACK ISSUES

FORMERLY 80-U.S.

May/Jun 1979
String packing techniques
Determine functions of the brain
How to win Nim-type games

Jul/Aug 1979
Create fast graphics
Renew lost programs
No-hardware lowercase mod.

Nov/Dec 1979
Function grapher/root finder
Home heat loss program
Restoring killed disk files

May/Jun 1980
Telecommunications with the TRS-80
BASIC game program technique
Produce sound with BASIC programs

Nov/Dec 1980
Simple payroll program
Digital plotter interface
Produce keyboard typeahead

May/Jun 1981
Line packing techniques
How to use "PRINT USING"
Animation and the TRS-80

Jul/Aug 1981
Descending lower case for Model I
Student timetable program
Easy tape loading for Model I

Sep/Oct 1981
Keyword search database
Compute a retail installment contract
The vertical mill: a 3-D plotter?

Nov/Dec 1981
Comparison shopper program
Real time clock construction
Pocket Computer biorhythms

Jan 1982
Microcomputers in business

Feb 1982
Micros and word processing

Mar 1982
Microcomputers and medicine

Apr 1982
Microcomputers and investments

May 1982
Space: An infinite frontier

Jun 1982
Games issue

Jul 1982
The TRS-80 in law offices

Aug 1982
Microcomputers in education

Sep 1982
Graphing and graphics

Oct 1982
Microcomputing tips and tricks

Nov 1982
Telecommunicate with your computer

Dec 1982
Disks and DOSs

Feb 1983
Computer languages: Reports on Pascal, COBOL, Pilot and Forth

Mar 1983
Data base management

Apr 1983
Computers in government

May 1983
Special peripherals issue

Jun 1983
Low-cost changes for your computer

Cost is $4.00 for each back issue ordered. Use the reply card to order, or you may write/phone Basic Computing, 3838 South Warner St., Tacoma, WA 98409-4698, (206) 475-2219.

Editorial

By Cameron C. Brown, Editor

The annual summer sale is now on. Prices are dropping so fast, most advertisers are replacing price quotes with a "call now" message. Some people have written to say how frustrated they are at having paid so much, way back then, compared to today's prices. Sorry, they don't have my sympathy.

I was one of those who purchased a Hewlett Packard HP-35 calculator for over $500. I knew the price would get down to a tenth of that, but a calculator in the future does little to solve a problem today. The same, "Let's wait", or "Why did we pay so much?" attitude exists in many places. Those who say such things forget that each year that goes by without a computer is one more year which can never be recaptured. For the offices, schools, or homes that can't afford adequate machines, they should at least consider the really inexpensive models. Even Radio Shack now offers an entry-level computer. The MC-10 is below $120 and may be less at some outlets.

It all points in one direction. The age of the disposable computer is almost here. I first heard that term used by my attorney, who had just purchased a Commodore for "her children." She was really using the machine to learn about those mysterious terms such as byte, CPU, and BASIC. Why pay thousands if you can get exposed to the wonders of computing for under $50? She does not expect this machine to be the one she finally uses when, and if, she ever chooses to computerize her office or home. But it is a machine for learning what a program is and what data is. Her attitude is to be commended.

Too many people look at the low-end computer market as a threat to their investment in more expensive machines. How silly. TRS-80 owners have a solid, but not too glamorous, machine. As more and more people are exposed to microcomputing, the numbers that join up with the TRS-80 line will continue to grow. That is to our benefit. More users means better software, more choices, and lower prices.

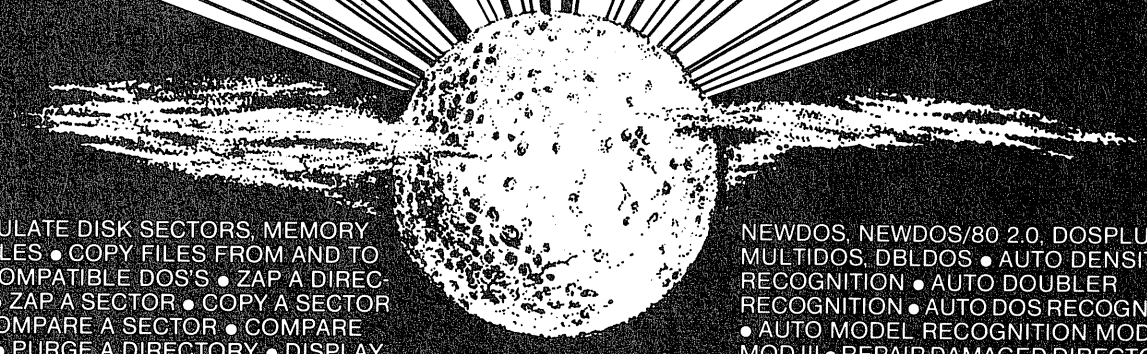
This issue is our first under the banner *Basic Computing*. Notice that we are still *The TRS-80 User Journal*. We thought long and hard and chose to stay associated with the TRS-80 line of microcomputers. We are convinced the future is rosy. Let everyone go out and buy a disposable computer. When the time comes for some practical computing, we will have a new reader.

EVERY FEW YEARS A UNIQUE PROGRAM ARRIVES

...SO PRACTICAL...SO USEFUL IT BECOMES HERALDED BY COMPUTER OWNERS INTERNATIONALLY AS A TRUE CLASSIC...AN INDUSTRY STANDARD...HIGHLY ACCLAIMED BY EVERY MAJOR TRS-80 PUBLICATION. YOU'VE SEEN IT WITH ONLY A HANDFUL OF TRS-80 PROGRAMS...NOW EXPERIENCE A LEGEND WITH THE NEW...

VERSION 3.0

SUPER UTILITY PLUS



MANIPULATE DISK SECTORS, MEMORY AND FILES • COPY FILES FROM AND TO NON-COMPATIBLE DOS'S • ZAP A DIRECTORY • ZAP A SECTOR • COPY A SECTOR AND COMPARE A SECTOR • COMPARE FILES • PURGE A DIRECTORY • DISPLAY DISK MAPS • EXAMINE FILE LOCATIONS • READ AN ALTER DATA ADDRESS MARKS • DOUBLE SIDED DRIVE SUPPORT LDOS, DOSPLUS, AND MULTIDOS • MORE • BOOTS ON MOD I OR MOD III 35, 40, OR 80 TRACKS • SINGLE OR DOUBLE DENSITY • TEST MEMORY • READS ALL CURRENT POPULAR DOS'S: TRDOS, LDOS,

80 Microcomputer Software Choice — **1st PLACE 80-MICRO 1982 UTILITY PROGRAM OF THE YEAR**

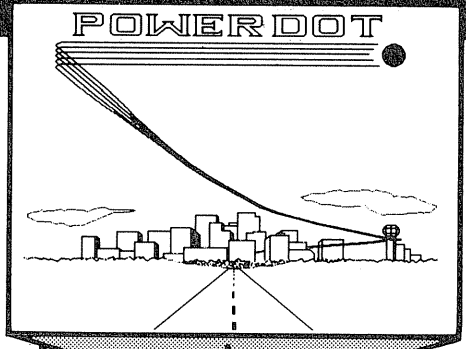
PROTECTED MEDIA
BACK UP COPY INCLUDED

NEWDOS, NEWDOS/80 2.0, DOSPLUS, MULTIDOS, DBLDOS • AUTO DENSITY RECOGNITION • AUTO DOUBLER RECOGNITION • AUTO DOS RECOGNITION • AUTO MODEL RECOGNITION MOD I OR MOD III • REPAIR DAMAGED DIRECTORIES • REPAIR BOOT SECTORS • REPAIR HIT & GAT TABLES • CREATE A SPECIAL OR CUSTOM FORMATTED DISK • REFORMAT WITHOUT ERASE • BUILT IN PRINTER ROUTINES • SOFTWARE BULK ERASE A DISKETTE • PATCHES TRSDOS AT OWNERS OPTION • EASILY CONFIGURED FOR ANY SYSTEM • MORE •

THE PROGRAM W/MANUAL 79.95 • THE TECH MANUAL 14.95 • THE BOOK INSIDE SU+ 19.95

POWERDOT

A GRAPHICS BREAKTHROUGH! Now the ULTIMATE in graphics design and editing, PowerDOT is the state-of-the-art in HI-RES Dot Mappers. Unmatched in quality and features, PowerDOT allows the user to create TRUE HI-RES printouts on the EPSON (Graftrax or Graftrax+ req.), or the C.Itoh PROWRITER from Leading Edge without ANY hardware modifications! The program has the ability to create drawings several times larger than the computer screen. You are only limited by disk storage. Your screen is merely a "picture window" to a much larger drawing area. You define the picture size. You select the print mode, hi-res, lo-res, or ultra-res. A very versatile utility and truly ahead of its time. Only **$49.95.**



POWERDRAW

A full screen graphics editor. PowerDRAW allows the user to draw directly to the screen utilizing cursor controlled movement. Graphics and text may be combined and saved directly to disk or tape, and recalled at a later time. The "SAVE" formats are: CONDENSED TOKENS, EDAS SOURCE, EDTASM SOURCE, BASIC DATA STATEMENTS, BASICS STRINGS, BASIC ARRAYS, and LOAD FILE FORMAT. The files may be run alone or merged into any type of program or application. Animation is also possible when screens are sequenced. Design custom forms, titles, pictures, and graphics for your applications. Received EXCELLENT reviews in INFOWORLD 6/82, 80-US 10/82, and 80-MICRO 10/82. Three printer drivers are included, and the program is JOYSTICK compatible. A GREAT value at only **$39.95.**

POWERMAIL

The most powerful, mailing system for the TRS80, Powermail is a highly sophisticated mass mailing system designed to run under all of the popular DOS's currently available for the Mod I or III. The program is written entirely in machine language for maximum operation speed, and occupies only 4K of the available RAM in your computer. There are no slow periods when Powermail is running. New features have been added to the program that others have always lacked. You now have the ability to keep track of mailings using the 24 flags that are incorporated into the Powermail program. The Powermail system will handle a file up to 8 megabytes, or 65,535 names, whichever is smaller. The program will run in as little as 32K and one disk drive, although 48K and 2 drives are desirable. The program will also sort the entire maximum file size and open up to 168 files simultaneously during the process. Only **$99.95**

PROGRAMS OF TOMORROW HERE TODAY

POWERDRIVER

New printer drivers for the new Radio Shack™ Superscripsit™. This new generation of custom printer drivers allows you to utilize all of SUPERSCRIPSIT'S features with your EPSON, PROWRITER and C.ITOH F-10 STARWRITER printers. You can now utilize all of the printers custom features such as compressed, expanded and proportional print, underlining, boldface, superscripting, and sub-scripting. All drivers can be called from within SUPERSCRIPSIT at "document open time." Order by printer designation as follows: Only **$29.95** each

POWERDRIVER E	EPSON MX70/80/100
POWERDRIVER P	C.ITOH PROWRITER
POWERDRIVER F	C.ITOH F-10

POWERSOFT
Products from Breeze/QSD, Inc.

Available from Selected Dealers Everywhere

11500 Stemmons Fwy., Suite 125
Dallas, Texas 75229
To order call toll free 1-800-527-7432
For product information (214) 484-2976

Letters to the editor

By Cameron C. Brown, Editor

My May issue arrived today, was received with much delight, and lo! there are the corrections to the fabulous Datex program from Paul Emmons (March 1983 issue).

I took the trouble to list Mr. Emmons program with Lister from Bullet80 in Connecticut and studied the details a little. It is a really beautiful job of engineering; it is remarkably fast, runs without a single on-error line, rejects erroneous inputs, and best of all, does not seem to lock up for garbage collection as do most BASIC data base systems. I noticed that the MID$ function was used freely throughout, as well as INSTR; both of which are little-used speedsters. This is the calibre of program seldom seen for free, it's better than the last few that I bought! I can truthfully say that it is the first such program I have not made major changes in.

In addition, it allows a range of options which is just fantastic for such compact code. I have never seen so many parameters, variable at the user's whim, without a complicated nest of linked programs. It is almost bullet proof; there are things you can do to foul it up, but it takes skill! I typed it in in one sitting, and of course made a myriad of typing errors.

In spite of the fact that I have a number of DBM and mail list systems, Mr. Emmons' comments are particularly apt. They all suffer from the shortcomings he lists. I think he must have spent a great deal of time on design before coding a line. It is as smooth as cream, and it does some things I did not think a simple basic system could do.

Thanks for the great magazine, and keep it coming. There are lots of us who need more than simple game listings. The Datexs (Datexes?) are real ego boosters.

Wesley J. Haywood, Jr.
Harvard, MA

Thanks for giving our ego a boost as well, but the next reader has a different view. One small point, the program wasn't free, your subscription helps pay for it. --Ed.

Your editorial in the April issue of *80-U.S. Journal* was quite interesting and somewhat challenging as well. I do not consider myself a skilled programmer in BASIC, but I do have several years of experience on which to draw. Also, I have just recently had the successful experience of converting the mailing list program in Bill Barden's book *Business Programming Applications* to run on the TRS-80 Color Computer. Clearly I was set up for a grand fall. The article on Datex in the March issue provided the opportunity for my tumble.

I was (and still am) interested in a data base program for the Color Computer. The reviews on Personafile and Color File have not stirred me to buy them. The Datex program looked interesting and I decided that I would try to implement it on the Color Computer. I have now spent about seven hours keying in the program from your listing, and I still have ahead of me all of the PEEKs, POKEs, and CMD's that were in the program as well as the task of adding an appropriate sort routine.

Since I do not have experience in assembly language programming, the sort routine will have to be a BASIC slowpoke. I do feel discouraged at this point and this week's time for computer work is at an end. I feel like asking the question "Why don't you publish some decent programs for my Model XX computer?"

Some of your editorial suggestions of better commenting the machine-dependent programming techniques seem to be very appropriate. However, there is another approach that would encourage more efforts to adapt "foreign" programs. Make the Model III programs more available to the CoCo owners and vice versa. I think that I would have much preferred to have spent the seven hours in adapting Datex to run on the CoCo instead of keying in the program and still having real work ahead of me.

Adapting Model III programs to run on the Color Computer may be feasible, but some are more feasible than others. You encouraged us, your readers, to give it a try. I would ask you to at least make it a little easier for us.

E. Wade Miner
Springfield, VA

You should have looked before you leapt. But please send in the conversion when you are done, we would love to print it. --Ed.

While poring over my treasured back issues of *80-U.S. Journal*, I came across the Un-Number utility on page 32 of the April 1982 issue. That program re-numbered the lines of a BASIC program so they were all zero.

Remembering that Microsoft BASIC allows a linenumber no higher than 65529, I made some modifications to Mr. Causer's utility. It will now run on the Model III as well as the Model I. The new version protects the program from a person of questionable character by preventing your own REMarks from being deleted. Append the utility to your program, add your remarks in lines 65521 to 65529 (you can change the lower boundary), and RUN 60000. Your REMs will be safe, undeleted, and numbered 65530.

You publish a fine magazine.

David Lewis
Shady, NY

The changes Mr. Lewis suggests you make to Mr. Causer's program are:

```
60000 P = PEEK(16548) + PEEK(16549)*256
60010 L=PEEK(P+2)+PEEK(P+3)*256: PRINT L: IF L>65520 THEN POKE P+2,250: IF L=65529 THEN END
60015 P=PEEK(P+0)+PEEK(P+1)*256: GOTO 60010
60017 REM THIS PROGRAM TAKES REM STATEMENTS NUMBERED FROM 65521 TO 65529 AND RENUMBERS THEM ALL TO 65530. BY DAVID LEWIS, BOX 88, SHADY, NY 12479
60019 REM HAVE A DUMMY LINE 65529
```

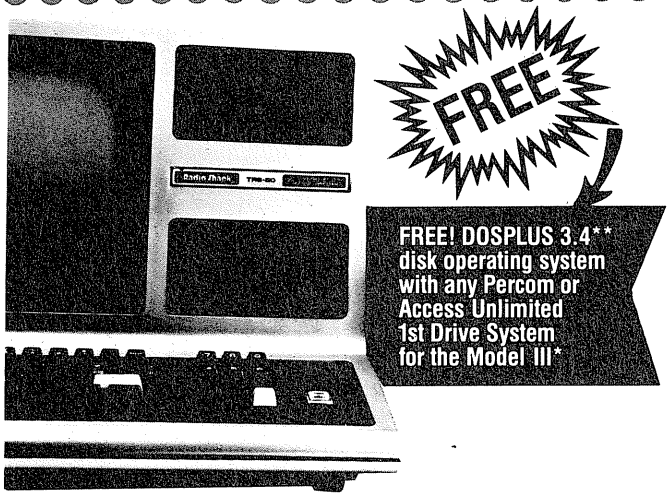
I can't remember a time, in my 59 years, in which I felt so attuned to something I was reading. I'm referring to your editorial in the March '83 issue. I felt as if someone with real talent was reading my thoughts. I experienced great relief just knowing that I was not paranoid and alone out in this jungle. I'm really fed up with the diatribe of your cross-country competitor and the insensitivity of Ft. Worth. A dozen letters to each has netted one reply. The phone numbers are useless. And forget about their invitations to drop in. I tried that once and developed a clinical depression. Recently, I have been soothing myself by looking and listening to the competition. I've allowed my fingers to run over strange keyboards without the anticipated nightmares of Benedict Arnold proportions. And, at the end of the day, I sit down with my Model III, SuperScripsit, DW II and fall in love with them all over again. It's a damn shame, isn't it?

My son has somewhat the same feelings about the CoCo. His instructors in college like to refer to our hardware as Trash 80's and are big Apple polishers. We still think we have them beat but we don't get much support.

I'm an old subscriber who dropped out a couple of years ago. I received a free issue in this month's mail and just signed on again. Keep up the good work and keep saying it like it is.

John E. Blank, M.D.
Sharon, PA

ACCESS UNLIMITED



SAVE $50.00 on a fully-tested PERCOM Disk Drive for Model III* Reg. $499.00. Now $449.00 ea or $750.00 Dual

Percom's internally-mounted drives, with their widely-acclaimed disk controllers, are completely pre-tested with a 48 hour Burn-in. So, you know it works right when you get it. Choose single or double-density storage capacity.

A First-Drive System includes the four-drive disk controller, one drive, power supply, mounting hardware, cables and a fully-illustrated, easy-to-understand owner's manual.

MAKE SURE IT'S DONE RIGHT! . . . Let Us Install Your First-Drive System

Purchase your Model III* Drives from us and for $79.95 plus shipping, you can have our experienced professionals perform a 48-hour burn-in of your Model III* computer, install the drive system, and check out your expanded system to make sure everything works correctly. Call for pricing of complete Model III* system with 2 to 4 Access Unlimited or Percom drives.

Get the best! Our own fully-tested drives* for as low as $275.00 Model I*

Every Access Unlimited Floppy Disk Drive is electrically and mechanically tested, then burned-in for 48 hours under operating conditions. The signed test list in your shipping carton is proof of that.

Our floppy disk drives offer you either single or double density. With double-density, it stores up to 364 Kbytes, depending on the format.

AFD40-1 SS/DD-**$275.00**
AFD40-2 Dual SS/DD-**$550.00**
TFD 44-1 DS/DD-**$350.00**
TFD 44-2 DS/DD-**$700.00**

"Percom" Quality for your "Model III"

Nothing but the best for your "Model III". Percom internally mounted drive systems: including 4 drive controller with gold edge connectors, double density disk drive or drives, all hardware and cabling. A free copy of DOS Plus 3.4 is also included with every first drive purchase.

TFD 340N1 one drive single sided double-density **$449.00**
TFD 340N2 two drive single sided double-density **$699.00**

Brand Spankin' New! SALE Dual Headed Drives for the Price of Flippies!

Now you can have a *dual headed "PERCOM" Drive System for your "Model III!!

TFD344N1 One drive dual headed double-density **$560.00**
TFD344N2 Two disk dual headed double-density **$860.00**

***completely compatible with programs existing on single sided or double sided diskettes.**

WE HAVEN'T FORGOTTEN THE "MODEL I"!

You have a good system and you just don't want to sell out to a Model III-Upgrade with the Percom Doubler II, the overwhelming favorite double density adapter for over **2 years!** Simply plug the adapter into your expansion interface and run either single or double-density programs. Comes complete with FREE DOS Plus 3.4 but will also run with LDOS NEW DOS 80 and TRS DOS.

Upgrade your Model I to Double Density — New Low Price $129.95
For "TRS 80 Model I & III" — Super Utility & V30 — **$55.45**

Percom Hard Disk — Quality for your System
*works with existing floppy drives
*can be daisy-chained up to four hard disks Reg. from $2495
***DOS Plus 4.0 or LDOS included FREE — Now from $1395.00**

Now available for immediate delivery in 5 & 10 megabyte configuration. Can be used for Model II, III*, "Apple II" or "IBM PC" computers.

SATISFIED GUARANTEED! If you're not completely satisfied with your Access Unlimited Disk Drive, return it within 15 days for a full refund. All our drives have a comprehensive 90-day limited warranty.

SPECIALS OF THE MONTH:

NEW "Star Gemini" Printer
Dot Matrix Model 10 — Reg. $499.00 Sale **$399.00**
100 CPS Model 15 — Reg. Retail $699.00 Sale **$699.00**

NEW "C-ITOH F-10 Starwriter" Printer
Reg. Retail $2295.00 Sale **$1595.90**

SPECIAL CLOSEOUT:

"Brother" HR1 Daisywheel Printer
Serial — Closeout Sale **$799.00**
Parallel — Closeout Sale **$745.00**

"DIABLO" MODEL 2300 MATRIX PRINTER
***Close Out Special* —**
• Rugged, commercial duty • 7x9 dot matrix • High Speed! 200 CPS!
• Top of the line, highest quality
Reg. Retail Price $2495.00 **CALL FOR OUR SPECIAL PRICE**

"SILENT SCRIBE" QUIET MATRIX IMPACT PRINTERS
DP-9000A (80 Col, 150 CPS, Graphix™ PLUS) ONLY **$1625.00**
DP-9501A (132 Col, 150 CPS, Graphix PLUS) ONLY **$1625.00**
DP-9620A (132 Col, 200 CPS, Graphix PLUS, Enhanced "NLQ" (Print Quality) ONLY **$1745.00**

"MPI" GRAPHICS PRINTER
• 150 CPS Sale, Only **$1099.00**

"TRANSTAR" 130P DAISY WHEEL PRINTER
• 16 CPS • Permits full range of word processing functions: proportional spacing, superscript, subscript, underscoring AND true boldface!
Now Only **$875.00**

"TI 810" PRINTER-Close Out! Your Price — Only **$1200**

"SOUND-TRAP" — Give your Epson, Okidata NEC Star or other likesize printer a nice quiet place of its own.
Only **$109.00**

HOT OFF THE PRESSES:

Inside Personal Computer Disk Storage
THE Necessary Basic Book, ***NEW*** from "Percom" Only **$5.95**

the MICRO SHOPPING CENTER

WHILE THEY LAST!

"Epson 14 x 20" Reg. $795 — Your Price **$699.00**

Save $$ on our most popular items!

Percom Data Separator (reg. $29.95)	Now **$23.95**
Screens for Models I*, II*, III Green, Lt. Blue, Dark Blue, and Amber. Bronze for color video (reg. $24.95)	Now **$12.95**
Head Cleaning Kit (reg. $29.95)	Now **$19.95**
Drive Numbering Tabs, pkg. 0-3 (reg. $4.50)	Now **$ 3.95**
Flip-N-File 8" (reg. $54.95)	Now **$28.95**
NEW Style Smoke Grey File 5¼", Holds 75	Now **$24.00**

Introducing the **"Color Coder"** — A complete diskette filing system by "Color Kits" 5 different colored library cases — each case will store 10 diskettes —

Reg. $39.95 — **$19.95**/set

MEDIA FOR LESS

"SENTINAL" complete with hub rings & one year limited warranty.

Single sided/Single density 5¼"	**$18.70** bx of 10
Single sided/Double density 5¼"	**$20.70** bx of 10
Double sided/Double density 5¼"	**$27.80** bx of 10
Single sided/Double density 8"	**$29.70** bx of 10
Double sided/Double density 8"	**$38.70** bx of 10

BUY DISKETTES IN BULK *AND SAVE* $$$$$$$$$$$$$$$$
These prices good by the case only —

Single sided/Single density 5¼"	**$179.00** case of 100
Single sided/Double density 5¼"	**$190.00** case of 100
Double sided/Double density 5¼"	**$225.00** case of 100
Single sided/Double density 8"	**$280.00** case of 100
Double sided/Double density 8"	**$340.00** case of 100

"BASF" 5¼" Single Sided/Double Density
Lifetime Limited Warranty. Reg. $44.95 Now **$24.90** bx of 10
5¼" Double Sided/Double Density — Now **$34.95** bx of 10

8" "BASF" Diskettes Double Sided/Double Density — **$39.95**

LIBRARY CASES — Holds 10 Diskettes Sale **$2.95 ea.**

DOS + 3.4 Reg. $159.00 **Sale $29.95**

Anti-Static Mats — Colors: Russett, Blue & Gold, Natural Brown & Golden Brown.

3' x 5' — Reg. $75.60	Sale: **$57.00**
4' x 6' — Reg. $120.90	Sale: **$91.00**
4' x 8' — Reg. $161.30	Sale: **$122.00**

"PAGEMATE" Typing Easels — $14.95

SAVE on an "Arrick Quick Switch"
Changes a "TRS-80" printer port or a peripheral between computers instantly and easily. Available for "RS-232" and "Centronics." Plugs included.
Now from $99.95. Cables from $26.95

BEAUTIFUL CUSTOM FURNITURE

"Atlantic Cabinet" — Oak!

50" Split Level Desk	**$249.00**
50" Personal Computer Desk	**$126.00**
38" Personal Computer Desk	**$105.00**
Matching Printer Stand	**$89.00**

"Toor" & "O'Sullivan" Computer Furniture Also Available — Call For Details, Prices

Print Wheels & Thimbles — From **$8.50** each

Printer Ribbons — For Centronics, Diablo, C-Itoh, Star and Epson 80, 100 From **$5.95** each

• LIMITED TIME OFFER/LIMITED QUANTITIES •
• Prices subject to change without notice •

* Trademark of Tandy Radio Shack Corp.
" " Reg. Trademarks • Prices do not include state taxes.

VISA **1 (800) 527-3475** master charge

Order by phone or by mail. We accept Visa, MasterCard, cashier's checks, certified checks, and money orders. With personal checks, allow additional time for bank clearance. Your bankcard will not be charged until your order is shipped. On orders over $1,000, we pay freight (surface only) and insurance; please add $3.00 shipping and handling over 50 lbs. Over 50 lbs., add $5.00 for orders under $1,000.00. Texas residents add 5% sales tax. Allow 2 to 4 weeks for delivery.

☐ Please send me a FREE catalog. I'm not ready to order at this time.
☐ YES, I'm taking advantage of your Sales prices.

Name _____
Company Name _____
Address _____
City _____ State _____ Zip _____
Phone Number ( ) _____

Quantity	Item	Unit Price	Subtotal
		Subtotal	
		State Sales Tax (Texas residents only)	
		handling charge	
		Total	

Check one:
☐ payment enclosed ☐ Visa ☐ MasterCard*

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

*If MasterCard, numbers above name: ☐☐☐☐

Expiration Date: ☐☐ — ☐☐

Authorized signature, if charged

ACCESS UNLIMITED

DEPT. C6/401 N. Central Expwy./Richardson, Texas 75080
Tel. 1-800/527-3475 214/340-5366
214/690-0207 — Sat. and Evenings Only

Notes, etc.

By Cameron C. Brown

Our New Name

It is now a fact. *80-U.S. Journal* is now named *Basic Computing*. We are still *The TRS-80 User Journal*. There are no plans to change that. Everyone here uses TRS-80's. We like them and see a long and healthy future for the Tandy line of microcomputers.

We do plan on including non-BASIC programs as well. Assembly language programs are always welcome, but lately, few have been submitted. Programs in other high-level languages such as Pascal will also be printed if they can appeal to a reasonably-sized part of our audience.

Model 12 Click

Pete Carr, of Port Orange FL, wrote to tell us about a way to tone down the Model 12 keyboard beep. It is so loud, most users disable it by entering CLICK OFF. Here is a simple fix. Underneath the main video and disk drive box of the Model 12 is a small, 1/4 inch diameter, hole on the front left side. Get a paper napkin and tear off enough to fill the hole. Don't pack it in, just fill it. This will filter the beep down to a tolerable level. Now you can have a faint and useful beep for feedback.

Underwater Enigma

Waldo T. Boyd of Geyserville, CA sent in the following bit of code from his Model II. It points out in a cute way that the TAB function really does print spaces when it is used. Changing the PRINTTAB commands to PRINT @ will help the "swimmer" keep his speed up.

```
1Ø CLS 'Once upon a time t
here was a
2Ø PRINT CHR$(Ø2) 'swimmer
 who liked
3Ø Y=RND(1Ø) 'to swim unde
rwater but
4Ø FOR A=1 TO 76 'each tim
e he head-
5Ø FOR X=1 TO Y 'ed for th
e far side
6Ø PRINTTAB(A)"." 'of the
lake some-
7Ø PRINTTAB(A+2)"o" 'thing
 seemed to
8Ø PRINTTAB(A+3)"." 'slow
his speed,
9Ø PRINTTAB(A+1)"o" 'and a
lmost held
1ØØ NEXT X 'him to a stand
still. All
11Ø NEXT A 'you could see
was bubble
12Ø PRINT 'trouble. Perha
ps you can
13Ø END 'discover what imp
eded him!!
```

Model III Cursor

This has been around for a while, but some newcomers to the Model III may not be aware of it. All you have to do to change the Model III cursor from blinking to solid is POKE 16412, 1. The cursor itself can be altered to a straight line by a POKE 16419, 95. If you would like to experiment with other cursors, try numbers other than 95. Choose from a list of ASCII character codes. You can look them up in Appendix C of the Model III BASIC Reference manual. Our thanks to Betty Fox of Gig Harbor, WA for sending in this tip.

Corrections

The table in our June article, 19 Color Computer Games, page 19, was missing its column headings. They are: 1) enjoyability, 2) graphics, 3) sound, 4) fun, 5) number of players, 6) complexity, and 7) requirements. Perhaps we should have made the table that month's puzzler.

Operation Genius, May 1983, was missing a lot of code. The lines for saving and loading from cassette were left out. We are relisting all lines that need altering. In many cases the change is minor, but necessary. Please make the following changes:

Delete line 780

```
820 IF (NU<58 AND NU>47) OR
NU=89 OR NU=78 OR NU=83 OR
NU=80 OR NU=67 THEN 830
ELSE 770
920 PRINT @0, CHR$(255);: PRINT
@0, "WHERE IS THE DATA?
DRIVE (0-3) OR (C)ASSETTE? ";:
NN=1: GOSUB 760: PRINT @0,
CHR$(255);: PRINT @0, "READ-
ING DATA . . . ";: IF NW$="C"
THEN 926
922 OPEN "I",1, "OPGEN3/DAT:
"+NW$: INPUT #1,G,RD$: CLOSE
1
924 GOTO 930
926 INPUT #-1, G, RD$
960 CLS: PN$="": FOR N=1 TO 49:
PN$=PN$ + STR$((P(N)): NEXT:
QN$= CHR$(34)+PN$+CHR$(34):
PRINT "WHERE SHOULD THE
DATA GO? DRIVE (0-3) OR
(C)ASSETTE? ";: NN=1: GOSUB
760: SD$=NW$: IF SD$<>"C"
THEN 970
962 PRINT: INPUT "PRESS
ENTER WHEN CASSETTE IS
READY"; E$
964 PRINT #-1, G; ","; QN$: GOTO
980
970 PRINT: INPUT "MAKE SURE
```

DESTINATION DRIVE IS NOT WRITE PROTECTED AND HAS AT LEAST 1 GRANULE OF SPACE ON IT. WHEN READY, PRESS ENTER"; E$: OPEN "O", 1, "OPGEN3/DAT:"+SD$: PRINT #1,G; ",";QN$: CLOSE 1
980 PRINT"WOULD YOU LIKE TO VERIFY THE DATA? ";: NN=1: GOSUB 760: IF NW$="N" THEN GOTO 590 ELSE IF NW$<>"Y" THEN 980 ELSE IF SD$<>"C" THEN 990
982 PRINT: INPUT "REWIND AND READY CASSETTE; THEN PRESS ENTER"; E$
984 INPUT #-1, V, RD$: GOTO 1000
990 PRINT: INPUT "READY DISKETTE IN DRIVE AND PRESS ENTER"; E$: OPEN "I", 1, "OPGEN3/DAT:"+SD$: INPUT #1, V, RD$: CLOSE 1

The Operation Genius game makes use of the reserved word NAME for execution of its machine language routines. If that reserved word has been implemented in your model (e.g. a double-density Model I), then it should be replaced with some other unimplemented or unused reserved word, such as LINE or POSN. The NAME reference is used in lines 640, 700, 830, 870, 890, 900, and 910 of the program.

Puzzler

The answer to our May question on recovering a Model II BASIC program can be found in the article by Brad Hoza that is on page 36 of this issue. The Color Computer questions on locking out the Break key (April) or implementing an AUTO command (May) are still unanswered.

This month we want to challenge the mathematical hobbyists in our audience. A "round number" is one whose binary representation has as many ones as zeroes. For example, nine is a round number since, in binary, it is 1001. The number eleven (1011) is not round. Our question is: How many round numbers are there that are less than or equal to the number 1000? Do not send us the program, just put the answer on a postcard and send it to July Puzzler, c/o *Basic Computing*, 3838 So. Warner, Tacoma, WA 98409. The earliest correct answer wins $10 and a tour of our facilities. We look forward to hearing from you.

Guitar seminar

Play and practice with your TRS-80

Models I/III

Lynard Barnes, Chicago, IL

The checkbook is balanced, the lunar lander has landed, the galatic empire has been saved, and the breadth and depth of machine language programming has crystalized on a shaky plateau of comprehension. If it isn't time for a two-week vacation, then it must be time for a guitar seminar. Pluck! Pluck!

Few musical instruments lend themselves to as much impassioned abuse as the guitar. Yet it remains, relatively speaking, unabusive to the ear. Some, like myself, whack on the thing for a lifetime and never acquire the exalted skills of Flamenco. With genuine, conscientious effort, a rhythmic, melodious strumming can be imitated. With a little more effort, something resembling a song can be produced. The bottom line is fun, and just plain selfish enjoyment. To keep fun and effort in near harmony, I've written "Guitar Seminar."

This program will display a total of one hundred ninety-six guitar chords. Not included in the count, are the ten chords which you may create yourself.

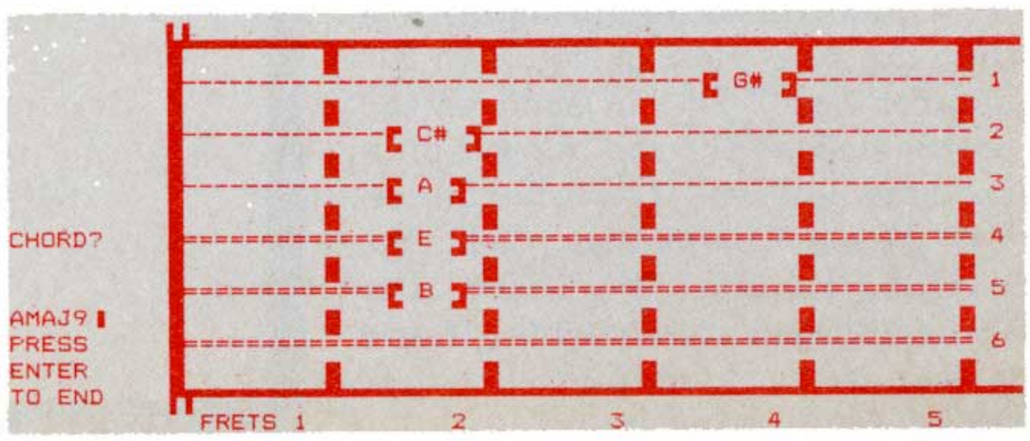
The purpose of the program is not to merely display a guitar chord. If you have ever encountered a series of difficult and/or unfamiliar chords, succeeding each other in a song you *really* wanted to learn to play, this program will provide you with some help. You may assemble up to twenty-five chords, including those you create, for playback in a repeatable, timed sequence. Even if you are totally unfamiliar with the guitar, a fifteen-minute session using the "P" option in this program will turn you into a person not totally unfamiliar with it. You will reap even greater benefits if you have a guitar!

Coded Data

The heart of this program is the C$ array, which contains the coded information for making up individual chords. The concept for employing this method to the complex problem of graphically presenting a guitar string depressed (or not depressed) at a particular fret, arose from the simple example of binary numbers. Only, in this case, the number of off-on states which may exist is a six by five matrix (five frets and six strings). Six decimal numbers are used to indicate column grid positions. Row numbers are determined in lines 500 and 530 of the program.

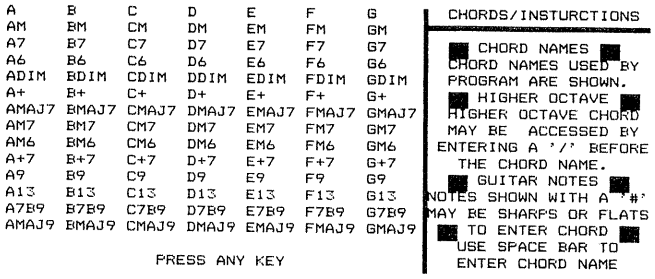
The chord Amaj9 (graphically spelled out in Figure 1) is represented by the number 922224. In lines 480 to 560 of the program, this number is peeled off (one digit at a time) from right to left. The PRINT@ position on the video monitor is held in the S(6,4) array, and the note corresponding to that fret position on the guitar is in the N$(6,5) array. The variable N is counted down, to indicate the string while each digit of the chord number is equated as a numeric value "U," which indicates the fret, or column number.

Figure 1: The program graphically displays one hundred ninety-six chords, showing where the fingers are to be placed on the frets of the guitar to make a chord.



The key to this technique is that the information contained in the position array must agree with information held in the musical notation array (G#, C, B notes, etc.). With this prerequisite, almost any musical instrument may be graphically represented on the TRS-80. There is one potential complicating factor, however.

Figure 2: The names of the ninety-eight basic chords which the program will display are shown. This screen is presented when the "I" command is selected from the options table.



Notice (in Figure 2) that if the first letter is removed from each of the chord names, the remaining information becomes exactly identical in each column. This is not happenstance.

When the name of a chord is entered, it is (in effect) divided into two portions. The R$ variable is the letter designation of the chord, only if the simple chords (A through G) are to be displayed. Otherwise, R$ indicates subcategories such as dim, +, and maj9. At all times, the letter name of the chord is translated into a numerical value represented by the variable R. This is achieved in lines 170 and 180. Afterward, the subroutine starting at line 660 is called and a branch is made on the basis of the numerical value of R.

Using Amaj9 as an example, R would equal zero (0) at line 620. Thus, at line 620, the program goes to line 630 where the first test fails (R$ is equal to maj9) and the LOOP counter is incremented until it reaches thirteen which is also the O2$ array position containing maj9. The numerical representation of the chord is picked up, a return made, and Amaj9 is displayed. A simple, efficient procedure. Whether the chord is Cmaj9 or Gmaj9, that procedure would not vary, except for the branch made at line 620 (resulting from the value of R). Score one for programmers.

Suppose we decide that a B seventh flatted ninth (B7B9 in Figure 2) is a remarkably rare guitar chord to encounter in a song. We desire a simple B flat. In this program, the only way to handle this is to add a fifteenth position to the O2$ array, and some additional data, to numercially represent a B flat and an A flat, and so on. This program modification is easy, though memory-consuming.

It is possible to have a B flat chord in the program without also including an A flat, a C flat and all of the rest. To do so, however, would give up one of the memory-saving features of the program presented here — cross indexing. Though there are ninety-eight separate chords, the effective names of those chords are all contained in an array, with a depth of only 14 positions (O2$(13)). Including one subcategory out of another would mean assigning an array to hold the names of each chord. A matter of individual programming taste, obviously, but memory considerations do play a part.

Pianos and Ukuleles

Since writing this program, I've enjoyed my guitar even more, and expanded my chord range beyond the "easy" ones. I have also expanded my interest in musical instruments. Yet, browsing through software catalogs, I am unable to find programs which give simple information on musical instruments. What I would really like to see is a program graphically presenting chord formations for the orchestra harp — harp as in Harpo, Chico and Groucho. My knowledge of this instrument is limited to Webster's cryptic "many strings." Many strings or not, I hope the techniques used in Guitar Seminar will give someone an idea to cover every string and, of course, how it should be plucked.

Program Listing for Guitar Seminar

```
10 CLS:CLEAR64:PRINT@342," GUITAR * SEMI
NAR ";:PRINT@466
,"CHORD SELECTION & PLAYBACK";:FORJ=0TO1
000:NEXT
100 CLEAR1000:DEFINTA-Z:MX=13:DEFSTRC:DI
MS(6,4),C$(195),
Q$(25),I#(25),O2$(13),N$(6,5):N0$=CHR$(1
76)+CHR$(149):
FORJ=0TO9:READA:NEXT:DATA9,8,7,6,0,1,2,3
,4,5:REM

OPTIONS
110 FORJ=0TO195:READC$(J):NEXT:FORJ=0TOM
X:READO2$(J):
NEXT:FORJ=0TO5:FORT=0TO5:READN$(J,T):NEX
T:NEXT:FORJ=0TO9:
READN(J):NEXT
120 FT=0:CLS:CD=CHR$(191):PRINT@202,STRI
NG$(44,176);:PRINT@270
,"I - CHORDS & GENERAL INSTRUCTIONS";:PR
INT@334
,"M - MUSIC NOTATION";:PRINT:PRINT@462
,"C - SINGLE CHORD DISPLAY";:PRINT@526
,"A - ASSEMBLE CHORDS FOR DISPLAY";
130 T=0:Q=0:PRINT@590,"P - DISPLAY ASSEM
BLED CHORDS";:PRINT@654
,"U - CREATE-A-CHORD";:PRINT@714,STRING$
(44,131);:
PRINT@93,"-OPTIONS-";:IFJC>=1,PRINT@0,"Y
OUR CHORD(S)";:FORJ=0TOJC:
PRINT@128+T,D$(J);:T=T+64:NEXT
140 GOSUB860:IFZ$="I",GOTO290ELSEIFZ$="C
",CLS:GOSUB580:
GOTO150ELSEIFZ$="U",GOTO330ELSEIFZ$="A",
GOTO200ELSE
IFZ$="M",GOTO900ELSEIFZ$="P",G=3:Q=KC:GO
```

```
TO25ØELSEGOTO14Ø:
REM

SINGLE CHORD DISPLAY
15Ø V=Ø:C="":O1$="":L=Ø:PRINT@512,"CHORD
?";:GOSUB6ØØ:
IFASC(O$)=13,GOTO12ØELSEGOSUB16Ø:IFV=9,G
OTO15ØELSEGOSUB73Ø:
GOTO15Ø
16Ø IFO1$=""ORO1$=" ",RETURNELSEIFASC(O1
$)=47,FT=3:RU=14:
GOSUB57Ø:O1$=RIGHT$(O1$,LEN(O1$)-1)ELSEI
FASC(O1$)<>47,FT=1:
RU=Ø:GOSUB57Ø
17Ø R=ASC(LEFT$(O1$,1))-65:IFR<ØORR>7,GO
TO72ØELSEIFLEN(O1$)>
=2,Y=-1ELSEY=Ø
18Ø R$=RIGHT$(O1$,LEN(O1$)+Y):IFLEFT$(R$
,1)="U",R=8
19Ø GOSUB62Ø:GOSUB48Ø:RETURN:REM

ASSEMBLE SONG CHORDS
2ØØ G=3:CLS:PRINT@32Ø,"CHORDS FOR DISPLA
Y/SONG":PRINT@448
,"YOU MAY ENTER A MAXIMUM OF 25 CHORD NA
MES.":GOSUB85Ø:CLS:
GOSUB58Ø
21Ø PRINT@512,"CHORD?";:PRINT@Ø,"RECORD"
;:PRINT@64,"PART I";
22Ø P$=CHR$(2ØØ):O1$="":V=Ø:GOSUB6ØØ:PRI
NT@768,P$;:
PRINT@832,P$;:PRINT@896,P$;:IFASC(O$)=13
,KC=Q-1:Q=KC:GOTO25Ø
ELSEC1=O1$:GOSUB16Ø:IFV=9,GOTO21ØELSEPRI
NT@32Ø,"PRESS";:
PRINT@384,"ENTER";:PRINT@448,"TO FILE";:
POKE16537,Ø
23Ø IFASC(C1)=47,C1="/"ELSEC1=""
24Ø B$=INKEY$:IFB$="",GOTO24ØELSEIFB$="
",GOSUB74Ø:GOTO21ØELSE
IFASC(B$)=13,Q$(Q)=C1+O1$:I#(Q)=VAL(C):Q
=Q+1:O1$="":PRINT@96Ø
,"-";Q;"-";:GOSUB74Ø:GOTO21ØELSEIFQ>=24,
KC=Q:GOTO25ØELSGOTO21Ø:
REM

DISPLAY ASSEMBLED/CREATED CHORDS
25Ø CLS:PRINTTAB(18)"SONG CHORDS - PART
II - PLAY":PRINT@32Ø
,"AUTO RUN THROUGH: HOW MANY TIMES";:INP
UTUA:IFUA=Ø,UA=1
26Ø PRINT"TIME DELAY (SECONDS): HOW MANY
```

```
:";:INPUTTD:GOSUB85Ø:
CLS:PRINT@192,"PLAY";:PRINT@64,"PART II"
;:PRINT@64Ø,"NAME";:
GOSUB58Ø:IFTD>=61,TD=6Ø*425ELSETD=TD*425
27Ø V=6:FORJ=ØTOQ:IFG=3,C=STR$(I#(J)):PR
INT@7Ø5,Q$(J);
ELSEC=STR$(B#(J)):PRINT@7Ø5,D$(J);
28Ø GOSUB48Ø:FORH=ØTOTD:NEXT:GOSUB74Ø:NE
XT:UA=UA-1:IFUA>=1,
GOTO27ØELSEW$="":GOTO12Ø:REM

I - CHORD NAMES
29Ø T=Ø:H=65:CLS:PRINT@44,"CHORDS/INSTUR
CTIONS";:PRINT@1Ø5
,STRING$(23,131);:J1=Ø:PRINT@172,CD;CD;"
 CHORD NAMES ";CD;CD:PRINT@236
,"CHORD NAMES USED BY";:FORJ=41TO1Ø21STE
P64:PRINT@J
,CHR$(17Ø);:NEXT:PRINT@3ØØ,"PROGRAM ARE
SHOWN.";
3ØØ PRINT@364,CD;CD;" HIGHER OCTAVE ";CD
;CD;:PRINT@428
,"HIGHER OCTAVE CHORD";:PRINT@492,"MAY B
E  ACCESSED BY";:
PRINT@555,"ENTERING A '/' BEFORE";:PRINT
@621,"THE CHORD NAME.";
:PRINT@684,CD;CD;" GUITAR NOTES ";CD;CD;
:PRINT@746
,"NOTES SHOWN WITH A '#'";
31Ø PRINT@81Ø,"MAY BE SHARPS OR FLATS.";
:PRINT@875
,CD;CD;" TO ENTER CHORD ";CD;CD;:PRINT@9
41,"USE SPACE BAR TO";:
PRINT@1ØØ5,"ENTER CHORD NAME";
32Ø FORJ=J1TO832+J2STEP64:PRINT@J,CHR$(H
);O2$(T);:T=T+1:NEXT:
H=H+1:T=Ø:IFH<=71,J1=J1+6:J2=J2+6:GOTO32
ØELSEGOSUB85Ø:GOTO12Ø:
REM

U - CREATE A CHORD
33Ø G=2:Q=Ø:UX=Ø:CLS:PRINT@148,"CREATE-A
-CHORD";:PRINT@32Ø,"
MAXIMUM: 1Ø
ALL SIX STRINGS MUST BE DESIGNATED AS 'S
TRIKE' (/) OR
'OPEN' (O).  TO END, TYPE '*'.":GOSUB85Ø
:UY=4
34Ø CLS:GOSUB58Ø:PRINT@Ø,"RT: >";:PRINT@
64,"LT: <";:
PRINT@128,"DN: +";:PRINT@192,"UP: L";:PR
INT@256,"OP: O";:
```

```
PRINT@320,"SK: /";:PRINT@384,"END: *";:T
A$="-":N=0:M=0
350 PRINT@S(N,0),STRING$(6,TA$);:UX=0:
PRINT@S(N,0)-4,"<";N$(N,0);">";:GOTO380
360 IFN<=2,TA$="-"ELSEIFN>=3,TA$="="
370 IFM=0GOTO350ELSEUX=-1:
PRINT@S(N,M+UX)-UY,CHR$(183);" ";N$(N,M)
;" ";CHR$(187);
380 T$=INKEY$:IFT$="",GOTO380ELSEPRINT@S
(N,M+UX)-4
,STRING$(6,TA$);:IFT$=".",GOTO390ELSEIFT
$=",",GOTO400ELSEIF
T$=";",GOTO410ELSEIFT$="L",GOTO420ELSEIF
T$="/",GOTO440ELSE
IFT$="*",Q=Q-1:GOTO470ELSEIFT$="O",M=0:G
OTO440
390 IFM<5,M=M+1:GOTO360
400 IFM>0,M=M-1:GOTO360
410 IFN<5,N=N+1:GOTO360
420 IFN>0,N=N-1:GOTO360
430 GOTO360
440 TC=TC+1:K1=M:O1$=STR$(M)+O1$:IFN<5AN
DT$="/"ORT$="O"ANDTC<6,
K2=M:GOTO410ELSEIFN>=5ANDTC<6,N=0:GOTO36
0
450 TC=0:U=VAL(LEFT$(O1$,2)):O$="U"+RIGH
T$(STR$(Q),1):
IFU=0,O1$="8"+O1$
460 M=0:O$=LEFT$(N$(N,U),1)+O$:D$(Q)=O$:
B#(Q)=VAL(O1$):O1$="":
PRINT@832,"NAME: ";D$(Q);:Q=Q+1:PRINT@70
4,"#: ";Q;:IFQ>=11,
Q=Q-1:GOTO470ELSEN=0:M=0:GOTO410:REM

U - DISPLAY NAMES
470 JC=Q:CLS:PRINT@128,"CREATED";JC+1;"
CHORDS";".  NAMES FOLLOW:"
:PRINT@256,"";:T=0:FORJ=0TOQ:T=T+1:IFT=8
,
PRINT:T=0:NEXTELSEPRINTD$(J);"  ";:NEXT:
GOSUB850:CLS:GOTO250:
REM

DISPLAY CHORDS
480 IFV=9,RETURNELSET#=VAL(C):C=STR$(T#)
:C=RIGHT$(C,7):
IFLEFT$(C$,2)=" 7"ANDFT<>10,C=" "+RIGHT$
(C,6):FT=5:GOSUB570:
GOTO520
490 IFLEFT$(C,2)=" 8",C=" "+RIGHT$(C,6):
GOTO520
500 C1=LEFT$(Q$(J),1):IFV=6ANDG=3ANDC1="
```

```
/",RU=14:FT=3:
GOSUB570ELSEIFG=3ANDC1<>"/",RU=Ø:FT=1:GO
SUB570
510 C1=LEFT$(D$(J),1):IFV=6ANDG=2ANDC1="
/",RU=14:FT=3:
GOSUB570ELSEIFG=2ANDC1<>"/",RU=Ø:FT=1:GO
SUB570
520 N=5:FORM=2TO7:
A$=MID$(C,M,1):U=VAL(A$):IFU=ØPRINT@S(N,
Ø)-6,"<";N$(N,Ø);">";:
N=N-1:GOTO560ELSEIFU=9,N=N-1:GOTO560ELSE
IFU=7
ORU=8THENGOTO560
530 IFU>=6,U=5
540 O$=N$(N,U):IFFT>=7,O$="X"
550 PRINT@S(N,U-1)-3,CHR$(183);" ";O$;"
";CHR$(187);:N=N-1
560 NEXT:O$="":V$="":RETURN:REM

FRET #
570 FORT=ØTO4:PRINT@S(5,T)+192,FT;:FT=FT
+1:NEXT:RETURN:REM

NECK
580 FF=Ø:T=Ø:F=Ø:TB$=STRING$(52,131):T$=
STRING$(52,176):PRINT@
1Ø,CHR$(181);CHR$(181);T$;:PRINT@9Ø6,CHR
$(151);CHR$(151);TB$;:
FORJ=74TO895STEP64:PRINT@J,CHR$(191);:NE
XT:FORJ=1ØTO54STEP1Ø:
FORO=74TO895STEP64:PRINT@O+J,CHR$(191);:
NEXT:NEXT:
FORJ=139TO8ØØSTEP128
590 GOSUB840:PRINT@J,STRING$(53,SX$);:S(
T,Ø)=J+6:T=T+1:NEXT:
FORJ=ØTO5:B=S(J,Ø):FORT=ØTO4:S(J,T)=B+F:
F=F+1Ø:NEXT:F=Ø:NEXT:
X1=1:FORJ=ØTO5:PRINT@S(J,4)+4,X1;:X1=X1+
1:NEXT:PRINT@972
,"FRETS";:F=1:FORT=ØTO4:PRINT@S(5,T)+192
,FT+F;:F=F+1:NEXT:
RETURN:REM
INPUT
6ØØ PRINT@768,"PRESS";:PRINT@832,"ENTER"
;:PRINT@896,"TO END";
610 PRINT@7Ø4+V,CHR$(138);:O$=INKEY$:IFO
$="",GOTO61ØELSE
IFO$=" ",RETURNELSEIFASC(O$)=13,RETURNEL
SEL=L+1:
V=V+1:O1$=O1$+O$:PRINT@7Ø4,O1$;:GOTO61Ø:
REM
```

```
GET STR$
62Ø IFR>=8,GOTO7ØØELSEFORJ=ØTOMX:
ONR+1GOTO63Ø,64Ø,65Ø,66Ø,67Ø,68Ø,69Ø
63Ø IFR$="A",C=C(14Ø+RU):RETURNELSEIFR$=
O2$(J),
C=C(J+14Ø+RU):RETURNELSEIFJ<MX,NEXTELSEG
OTO7ØØ
64Ø IFR$="B",C=C(168+RU):RETURNELSEIFR$=
O2$(J),
C=C(J+168+RU):RETURNELSEIFJ<MX,NEXTELSEG
OTO7ØØ
65Ø IFR$="C",C=C(Ø+RU):RETURNELSEIFR$=O2
$(J),
C=C(J+RU):RETURNELSEIFJ<MX,NEXTELSEGOTO7
ØØ
66Ø IFR$="D",C=C(28+RU):RETURNELSEIFR$=O
2$(J),
C=C(J+28+RU):RETURNELSEIFJ<MX,NEXTELSEGO
TO7ØØ
67Ø IFR$="E",C=C(56+RU):RETURNELSEIFR$=O
2$(J),
C=C(J+56+RU):RETURNELSEIFJ<MX,NEXTELSEGO
TO7ØØ
68Ø IFR$="F",C=C(84+RU):RETURNELSEIFR$=O
2$(J),
C=C(J+84+RU):RETURNELSEIFJ<MX,NEXTELSEGO
TO7ØØ
69Ø IFR$="G",C=C(112+RU):RETURNELSEIFR$=
O2$(J),
C=C(J+112+RU):RETURNELSEIFJ<MX,NEXTELSEG
OTO7ØØ
7ØØ FORJ=ØTOKC:IFR$=RIGHT$(Q$(J),2),C=ST
R$(I#(J)):RETURNELSENEXT
71Ø FORJ=ØTOJC:IFR$=RIGHT$(D$(J),2),C=ST
R$(B#(J)):RETURNELSENEXT
72Ø PRINT@448,"NO SUCH";:FORJ=ØTO5ØØ:NEX
T:GOSUB74Ø:V=9:RETURN:REM


CLEAR SIDE
73Ø OT$=INKEY$:IFOT$=""THENGOTO73Ø
74Ø P$=CHR$(2ØØ):PRINT@448,P$;:PRINT@576
,P$;:
PRINT@512,P$;:PRINT@Ø,P$;:PRINT@7Ø4,P$;:
PRINT@768,P$;:
PRINT@832,P$;:PRINT@896,P$;:PRINT@32Ø,P$
;:PRINT@384,P$;
75Ø FF=Ø:FORX=ØTO5:GOSUB84Ø:PRINT@S(X,Ø)
-6,STRING$(4,SX$);:
FORY=ØTO4:PRINT@S(X,Y)-4,STRING$(7,SX$);
:NEXT:NEXT:RETURN:REM


76Ø DATA
332Ø1Ø,991Ø1Ø,932315Ø,3Ø221Ø,9Ø1212,93211
Ø,935459,
335343,939243,912115Ø,332333,912215Ø,33232
9,932433,
993331,113321,993334,913333,9Ø2323,99433
2,993335,
993324,993323,994334,993534,996533,99343
4,933335:'   C
77Ø DATA
954232,993231,9ØØ212,924232,993434,99433
2,954222,
93Ø231,923239,934339,554555,923329,55454
9,8ØØØ222,
995554,335543,335353,935555,994545,99655
4,995557,
335343,995545,996556,995754,998755,99565
6,932422:'   D
78Ø DATA
9221ØØ,922ØØØ,8Ø2213Ø,8Ø2212Ø,992323,932
11Ø,992444,
992433,922Ø2Ø,93213Ø,922132,8Ø2Ø12Ø,9221
31,922142,
954232,993231,594535Ø,924232,993434,99433
2,957679,
557565,923235Ø,994536,554555,8Ø3443Ø,5545
4Ø,954655:'   E
79Ø DATA
133211,133111,131211,13Ø231,99Ø1Ø1,99322
1,9Ø321Ø,
131111,99Ø111,9Ø1221,933243,1Ø1239,93324
2,933253,
965343,994342,945349,945343,991212,99544
3,965333,
94Ø342,934345Ø,9Ø5647,665666,965668,66565
Ø,965766:'   F
8ØØ DATA
32ØØØ3,355333,32ØØØ1,992433,992323,99544
3,995432,
353333,392333,993443,323239,393459,99343
4,225231,
987565,996564,897869,993535,993434,99766
5,987555,
969564,956569,967669,933243,967769,93324
2,333555:'   G
81Ø DATA
8ØØ222Ø,8ØØ221Ø,8ØØ2223,8ØØ2222,991212,9
Ø3221,8ØØ2224,
8ØØ2213,8ØØ2212,9Ø3223,8ØØ2423,995422,99
2323,922224,
355433,355333,353433,359453,992323,9Ø544
3,8ØØ5432,
353333,392333,993443,7933243,393459,7933
242,225232:'   A
82Ø DATA
8Ø224442,224432,224242,924444,999191,321
999,924349,
224232,994434,921293,221222,921224,22121
9,921322,
992225,7355333,7353433,7359453,991212,99
3221,7995432,
```

```
7353333,594555,993223,992423,995422,9923
23,922224:'        B
83Ø DATA"",M,7,6,DIM,+,MAJ7,M7,M6,+7,9,1
3,7B9,MAJ9,
E,F,F#,G,G#,A,B,C,C#,D,D#,E,G,G#,A,A#,B,
C,
D,D#,E,F,F#,G,A,A#,B,C,C#,D,E,F,F#,G,G#,
A,
7Ø,68,66,71,69,69,67,65,7Ø,67
84Ø FF=FF+1:IFFF<=3,SX$="-":RETURNELSEIF
FF>3ANDFF<=6,SX$="=":
RETURNELSEIFFF>=7,FF=Ø:SX$="-":RETURN
85Ø PRINT@975,"PRESS ANY KEY";
86Ø Z$=INKEY$:IFZ$=""THEN86ØELSERETURN:R
EM

-M-
87Ø IF(Y=8ORY=6ORY=1ORY=3ORY=5),YØ=61ELS
EYØ=32:RETURN
88Ø IFY=Ø,X=5ELSEIFY=1,X=6ELSEIFY=2,X=7E
LSEIFY=3,X=8ELSEIFY=4,
X=9ELSEIFY=5,X=1ØELSEIFY=6,X=4ELSEIFY=7,
X=3ELSEIFY=8,X=2ELSE
IFY=9,X=1
89Ø RETURN:REM

MUSIC SCALE
9ØØ CLS:D=128:T2=576:OD$=STRING$(59,"=")
:T=Ø:
FORJ=64TOT2STEPD:PRINT@J+2,OD$;" ";CHR$(
N(T));:T=T+1:NEXT:
H=J:T=9:FORJ=64TOT2STEP64:
PRINT@J+2,CHR$(191);:PRINT@J+6Ø,CHR$(191
);:NEXT:
91Ø Q=Ø:FORX=H-64TO128STEP-D:PRINT@X,CHR
$(N(T));:T=T-1:NEXT:
PRINT@7Ø4,STRING$(63,14Ø);:T=4:FORJ=388T
O68STEP-64:I(Q)=J:
Q=Q+1:NEXT:I(6)=452:I(7)=516:I(8)=58Ø:I(
9)=644:Q=Ø:
H=Ø:N$=CHR$(186)+CHR$(187):PRINT@832,"MU
SICAL BAR SCALE";:
GOSUB85Ø:CLS:REM

STRINGS
92Ø PRINT@Ø,"GUITAR STRINGS & NOTES:":PR
INT@192,"
STRINGS LABLED 1 TO 6. FIVE FRETS (COLUM
NS) ARE SHOWN.
THE NOTES INDICATED ARE FOR REFERENCE PU
RPOSES ONLY."
93Ø T1=Ø:T2=1:J1=Ø:FORJ=648TO1Ø21STEP64:
PRINT@J-2,T2;:
FORT=ØTO5:PRINT@J+T1,CHR$(17Ø);N$(J1,T);
:T1=T1+8:NEXT:J1=J1+1:
T1=Ø:T2=T2+1:NEXT:GOSUB86Ø:GOTO12Ø
94Ø END
```

Colorful two-part harmony

POKE a polka on your Color Computer

Color Computer

Leo Christopherson, Tacoma, WA

This article describes a two-part music output routine for the Color Computer. If you type in and RUN the BASIC program listed with this article, you will be able to hear a sample musical selection. If you like what you hear, you can follow the directions to modify the program to play your own musical selection. Once the music plays properly, you can easily make it part of one of your own BASIC programs.

The Heart of the Matter

The 6809 music subroutine is what makes all of this possible. The subroutine outputs two square waves simultaneously. It uses the D/A converter as an output for the sum of the volumes of the two waves. The routine will play two pitches for a specified duration, or provide a rest (silence) for a specified duration. It is not necessary to understand the operation of the routine to use it. However, for the details, see the description of the 6809 routine accompanying this article.

Getting Up and Running

To begin, type the BASIC listing into your computer (minus REMark statements, if you wish). With the exception of REM lines, the program has to be exactly as shown! Be sure to save the program before you RUN it.

If all is well, when you RUN the program, you will hear the music play through the television speaker and then the machine will stop with a break in line 100. If this does not happen, load the saved program again and check it until you find your error. This program uses "string packing," a technique which imbeds the machine level code in a string in a line of the BASIC program. The DATA item numbers must be correct, and the number of periods in lines 1000, 1005, 1010, 1015, 1020, and 1025 must be at least as many as indicated. (A few more will not hurt anything, but a few less will likely crash the program.)

When the program has run successfully and the music plays correctly, you can DELete line 15, and lines 1998 to 10030, since once the lines are packed, they need not be packed again for the music to play.

Doing Your Own Thing

Let me describe how you can get the music routine to play your own music data. We'll make up a new Part (1) to illustrate. You may DELete everything *except* lines 10, 100, 1000, 1001, 1030, and 20000 to 20020. Be sure that you have RUN the program so that the string in line 1000 will be packed. (Don't worry about the strange look of the packed string when it is LISTed. It's okay that way. Be sure not to EDIT a packed string, by the way. You can destroy the effect of packing by EDITing.

Here's the first part for "Mary Had A Little Lamb" in the key of F: A2,F2, G2,C2, F2,A1, G2,C2, A2,F2, A2,F2, A2,F2. The "2" after the letter for the note means that the note is in the "middle C" octave. A "1" would be the octave below "middle C."

We have to put the music data into the form that the music routine expects. Each sequence of two notes consists of a duration followed by the two pitches. Each rest sequence consists of a "flag" of "254" followed by the duration of the rest. The entire part must end with a flag of "255." We'll use 64 as the duration. We find the pitches by looking them up in the "Music Data Number Pairs" table (Table 1). The pairs are arranged with the first note being the highest pitched of the two to be played. For the first note pair (A2,F2), we find A2 at the left of the table and follow across to the second note listings for F2. It happens to be the first listing and reads: F2:37,46. This means the two pitch numbers we need are 37 and 46. The data for Part (1) goes into lines 3000 to 3002 and looks like this:

```
3000 DATA 64, 37, 46, 64, 42, 62, 64, 47, 75, 64
3001 DATA 42, 62, 32, 37, 46, 254, 120, 32, 37, 46
3002 DATA 254, 120, 32, 37, 46, 254, 120, 255
```

Note the "255" at the end of the data. Add these lines to your program now.

Next, we need to provide a string into which we will pack our data. We need to know how long the string is to be. If you count the number of data items in lines 3000 to 3002, you will see that we need a string with a length of 28. We also need to know where the music-part string will start in terms of its absolute memory address in the computer. We use the VARPTR function to find this out. Type the following lines into your program:

```
1005 PART$(1)=".....(.28.PERIODS.)....."
1006 A1=PEEK(VARPTR(PART$(1))+2):
     A2=PEEK(VARPTR(PART$(1))+3)
```

Now we have to put in the part of the program that will pack PART$(1). Type in the following lines:

```
  15 GOSUB 10005
10005 A0=A1*256+A2: RESTORE: FORN=A0 TO
A0+27: READD: POKEN,D: NEXTN
10030 RETURN
```

The last thing to do is to tell the music routine where to find the data and then to call the routine. The music routine expects the most significant byte of the data string's starting address to be in zero (direct) page memory address 248. The least significant byte of the address needs to be in zero page 249. We use the EXEC statement to address the machine-level music routine whose absolute memory address is known by the variable "MUSIC." Add line 20 to your program:

```
20 POKE248,A1: POKE249,A2: EXEC MUSIC
```

Everything should be ready now, but (again) it's a good idea to save the program before RUNning it. When

you do RUN it, you should hear the first part of "Mary Had A Little Lamb."

Possible Problems

I have developed this sound routine for a program I am writing for Radio Shack. The music in that program is all in the keys of F major or B♭ major. Thus, I've only worked out a fairly small number of the note pairs possible within the limits of pitches available here. Table entries followed by a question mark are notes I didn't need to use at all.

What if you need a note pair not listed in the table? The number in parentheses, following the name of the first note in the table, is essentially the correct pitch to use. For example, we would expect B2 and G2 to be 32,41. However, the wider the pitch gap between the two notes to be played, the more likely they will be out of tune. Therefore, I have included the tuning routine in lines 20000 to 20020.

This tuning routine assumes that the music string in line 1000 has been packed. You must have a musical instrument available to act as a correct source of pitches. To use the routine, type GOTO 20000. At the INPUT question mark you see, enter the two pitch numbers you want to use. The routine will only let you hear the first of the two notes even though it's outputting both of them. Using your source of true pitches, check the first note. Vary the number of the first note until it's as close as possible to true pitch. After finding the pitch number for the first note, enter the notes with the second note first, and do the same testing for the second note.

Caution: Do not use a "34" as data. The BASIC program will not accept a "34" in your data strings. Unfortunately, the "34" is just at the pitch of B♭2. I got around it by using B♭2 with a pitch low enough that I needed a "35" instead of the "34" for the notes to be in tune.

If the instrument you're using varies a good deal from standard musical tuning, you may have to work out a whole new table. The table I've included does conform to three different instruments I've tried, so it should come close for you, too.

You're On Your Own, Now

Please feel free to use this music routine in your programs if you wish. And, if you come up with something in the way of an improvement, I would like to hear about it. Please address any comments to me in care of *Basic Computing*, 3838 South Warner Street, Tacoma, Washington 98409.

Table 1 — Music Data Number Pairs

1st Note		2nd Note(s)			
F3	(22)	C3:24,30	G2:24,40		
E3	(23)	?			
E♭3	(25)	?			
D3	(27)	F2:27,46			
D♭3	(28)	?			
C3	(30)	A2:30,36	E2:31,49		
B2	(32)	?			
B♭2	(34)	G2:35,41	D2:35,55		
A2	(36)	F2:37,46 C2:37,62	E2:37,49 F1:37,95	D2:37,55	D♭2:37,59
A♭2	(39)	?			
G2	(41)	E2:41,49 C2:42,62	E♭2:41,42 B♭1:42,70	D2:42,55 G1:42,84	D♭2:42,59 E1:42,101
G♭2	(44)	D2:44,56			
F2	(46)	D2:47,56 G1:47,84	D♭2:47,59 F1:47,95	C2:47,63	A1:47,75
E2	(49)	D2:50,56 G1:50,84	D♭2:50,59 C1:50,128	C2:50,63	B♭1:50,71
E♭2	(53)	C2:53,63	B♭1:53,71	A1:53,75	E♭1:53,107
D2	(56)	B1:56,67 F1:56,95	B♭1:56,71 E♭1:57,107	A1:56,75	G1:56,84
D♭2	(60)	A1:60,75	E1:60,101		
C2	(63)	B♭1:63,71 E1:63,101	A1:63,75 E♭1:64,107	G1:63,85 D1:63,114	F1:63,95 C1:63,128
B1	(67)	G1:67,85	F1:67,95		
B♭1	(71)	G1:71,85	F1:71,95	E1:71,101	D1:71,114
A1	(75)	F1:76,96	E1:76,102		
A♭1	(80)	?			
G1	(85)	F1:85,96	E1:85,102	C1:85,128	
G♭1	(90)	?			
F1	(96)	E♭1:96,108	D1:96,114	C1:96,129	
E1	(102)	C1:102,129			
E♭1	(108)	?			
D1	(114)	?			
D♭1	(121)	?			
C1	(129)	?			

Comments on the 6809 Routine

Steps 0 and 1 get the starting address of the music data to be played. This address is POKEd into zero page addresses 248 and 249 by the BASIC program. Steps 2 and 3 adjust the starting address back three bytes to fit the requirements of the music routine.

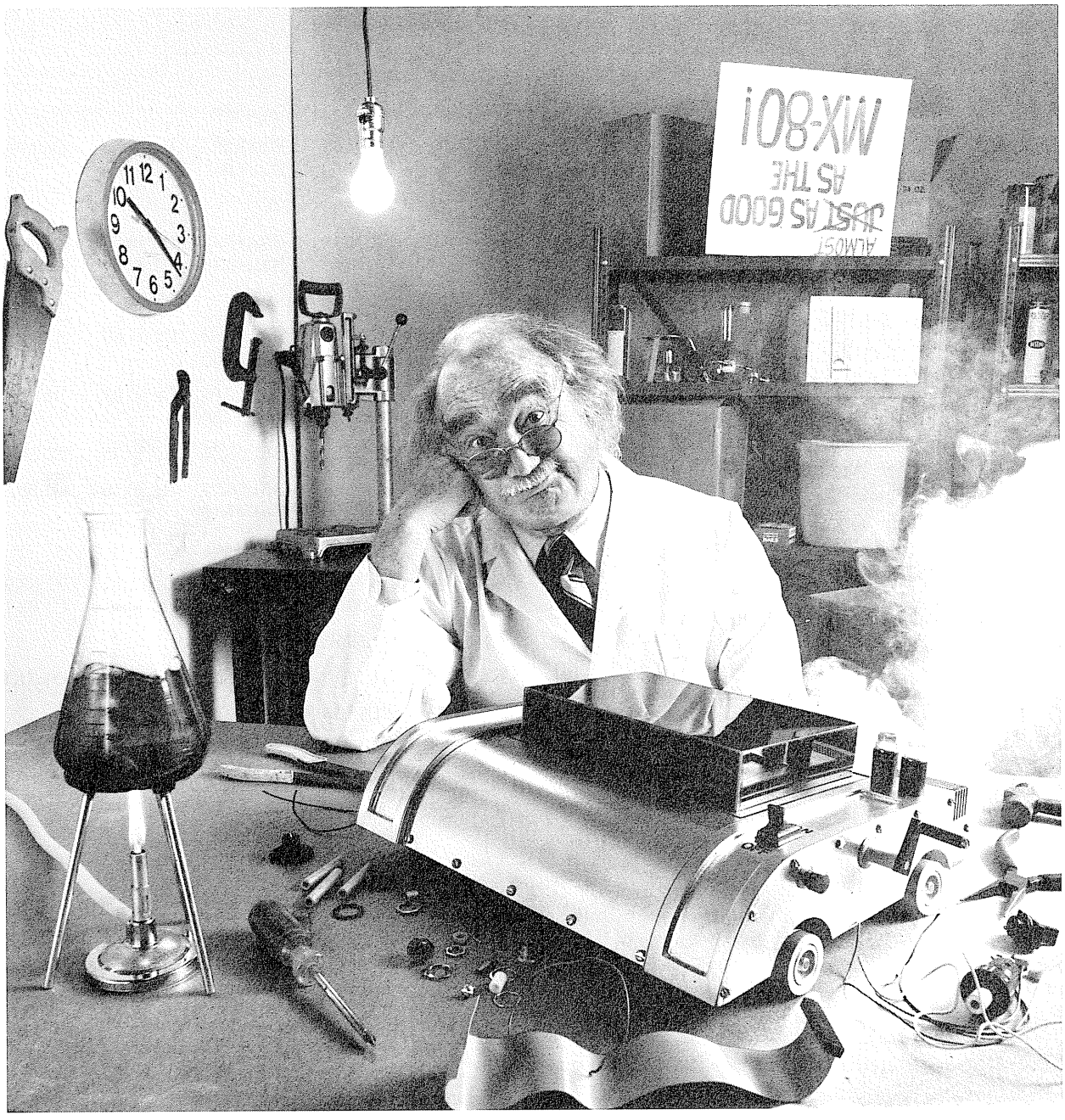
Steps 4 through 12 turn off the 60-cycle interrupt which can cause a "fuzzy" sound in the music when present.

Steps 13 through 39 enable the six-bit digital-to-analog sound output.

Steps 40 through 48 set the duration count in the X register to 1 and clear the output flags in zero page 245 and 247.

Steps 49 and 50 decrement the duration count. If the count is not zero, the program branches to step 100 to continue cycling the same note as before.

Steps 53 through 58 advance the data address three bytes and check the byte stored at the new address. If the byte is a 255, the music is done and the program stops by shutting off the six-bit sound and allowing the 60-cycle



For everyone who's tried to top the MX-80, bad news. We just did.

Epson.

The Epson MX-80 is the best-selling dot matrix impact printer in the world. It has been since its introduction. And despite the host of imitators it spawned, no one has been able to top it. Until now.

FX-80: Son of a legend.
The new Epson FX-80 is far more than just doo-dads added on to last year's model. It's the most astonishing collection of features ever assembled in a personal printer.

For starters, it's fast: 160 CPS. And clean. All the print quality Epson is famous for in a tack-sharp 9x9 matrix.

But that hardly scratches the surface.

Create your own alphabet.
With the new FX-80, you aren't limited to ASCII characters. You can create your own. Any character or symbol that can be defined in a 9x11 matrix can be added to the FX-80's already impressive library of type styles and stored in its integral 2K RAM.

So you can create "Sally's Gothic" or "Tom's Roman" just by downloading and modifying standard characters. Or you can create a custom set from scratch. Either way, you can store up to 256 new characters. And if you don't need a new alphabet, the RAM functions as a 2K data input buffer.

Who knows graphics better than Epson?
Nobody, that's who. And if you don't believe it, witness the FX-80.

With a 12K ROM capacity, the FX-80 gives you a few things the others don't. For example, not one, not two, but *seven* different dot addressable graphic modes are program selectable. And can be mixed in the same print line. Everything from 72 DPI (dots-per-inch) Plotter Graphics to the 640 dots per line resolution designed to match the remarkable monitor clarity of the Epson QX-10 personal computer.

And *that* is in addition to an astonishing array of 136 different user-selectable type styles including Proportional, Elite and Italic as well as the more conventional faces you get on other printers.

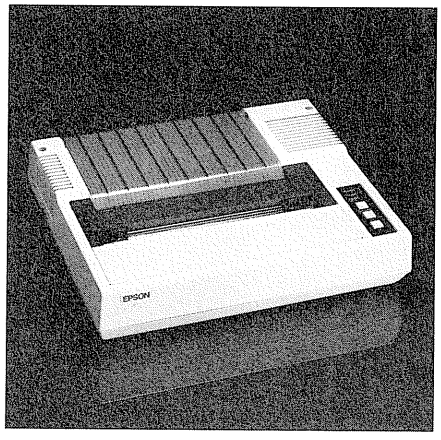
Hard-to-beat hardware.
The FX-80 has all the hardware features you've come to know and love on the MX Series: logic seeking, bidirectional printing, the by-now-famous disposable printhead, and more.

The FX-80 features an adjustable pin platen or optional friction/tractor feed, so you can use fanfold, roll or sheet paper . . . backwards or forwards. The FX-80 even gives you reverse paper feed.

And if you're printing forms, the FX-80 has a feature you're gonna love: a function that allows you to tear off the paper within one inch of the last print position.

Be the first on your block.
We'd be willing to bet that the FX-80 — like the MX-80 — will have its share of imitators. Don't be fooled. To make sure you get the genuine article, rush down to your local computer store right now and let them show you everything the FX-80 can do.

And while you're there . . . ask them to show you how it works with our computers.



EPSON

EPSON AMERICA, INC.

COMPUTER PRODUCTS DIVISION

3415 Kashiwa Street
Torrance, California 90505
(213) 539-9140.
Outside California, phone
(800) 421-5426 for the
Epson dealer nearest you.

interrupts again. This is done in steps 61 through 77.

Steps 78 through 91 find if the new data byte is 254. A 254 means no music is to be played. In music terms, a rest is required. The X register is loaded with the next two bytes and is then decremented. At the end of the delay, the data address is moved ahead two bytes and the program is cycled back to step 55 to test the next sequence of data. The data for a rest consists of two bytes: the 254 flag byte and the most significant byte of the delay count. The X register will get a least significant byte consisting of the first byte of the next sequence of data, but what it is doesn't really matter much.

Steps 92 through 99 load the counter in X with a new duration (the least significant byte of which is actually the first pitch byte). Then the pitch counters, zero page 244 and 246, are loaded with the two pitch counts.

Looping through the rest of the routine now amounts to decrementing each pitch counter and decrementing the duration counter. When the duration count is zero (steps 49 and 50) the program sets up for the next pair of notes. Otherwise, the routine continues to decrement the pitch counters in zero page addresses 244 and 246. When one of these counters gets to zero, its corresponding flag register (245 or 247) is toggled. That is, if it held a one, it becomes a zero and vice versa (the EORB$ instructions). The A register holds a 120 which is the output volume of the note. Then, depending upon whether the output flag is a zero or a one, the volume in A is either added or subtracted from the overall volume level in the output location $FF32 (65312 decimal).

Thus, the 6-bit sound output byte is caused to change in time with both pitches. Sometimes it contains zero. Sometimes it contains either a 120 or a 240.

6809 Machine Level Music Subroutine (BASIC Program Linenumber 1000)

Step Number	Decimal Code	6809 Instructions
0	222	LDU1
1	248	248
2	51	LEAU -3,U
3	93	PB
4	19	SYNC
5	134	LDA#
6	254	254
7	180	ANDA$
8	255	255
9	3	3
10	183	STA$
11	255	255
12	3	3
13	127	CLR$
14	255	255
15	32	32
16	134	LDA#
17	247	247
18	180	ANDA$
19	255	255
20	1	1
21	183	STA$
22	255	255
23	1	1
24	134	LDA#
25	247	247
26	180	ANDA$
27	255	255
28	3	3
29	183	STA$
30	255	255
31	3	3
32	134	LDA#
33	8	8
34	186	ORA$
35	255	255
36	35	35
37	183	STA$
38	255	255
39	35	35
40	79	CLRA
41	198	LDB#
42	1	1
43	31	TFR D,X
44	1	PB
45	151	STA$
46	245	245
47	151	STA$
48	247	247
49	48	LEAX -1,X
50	31	PB
51	38	BNE
52	47	STEP# 100
53	51	LEAU 3,U
54	67	PB
55	166	LDA ,U
56	196	PB
57	129	CMPA#
58	255	255
59	38	BNE
60	17	STEP# 78
61	134	LDA#
62	247	247
63	180	ANDA$
64	255	255
65	35	35
66	183	STA$
67	255	255
68	35	35
69	134	LDA#
70	1	1
71	186	ORA$
72	255	255
73	3	3
74	183	STA$
75	255	255

13 TIMES *FASTER* THAN dBASE II™

SUPER™

If you bought your computer to save time, then you need SUPER, the most powerful database system you can use. Power is a combination of speed, ease of use and versatility. SUPER has them all.

FAST – To demonstrate SUPER's speed, ISA retained a professional dBASE programmer to benchmark SUPER vs. the acknowledged leader. A simple mailing list application was chosen to minimize dBASE programming cost. The results:

Task	SUPER Time	dBASE II Time
Set up/Program	5:20	12:18:00
Input 100 records	50:29	1:27:50
Sort & Print Labels	6:41	4:18
Totals	1:02:30 hrs.	13:50:08 hrs.

Notice that SUPER was faster at every task where *your time* is involved—and saving your time is probably the whole reason you bought a computer.

EASY TO USE – SUPER won because of its ease of use. Since it is menu-driven, office personnel can easily learn to use SUPER to set up their own applications, speeding and simplifying dozens of tasks without the need of programmer support.

VERSATILE – SUPER, unlike other business programs, doesn't dictate how to run your business. With SUPER the computer does what *you* want, when *you* want, the way *you* want it. *SUPER may* **be the only business program you'll ever need**. It can handle customer files, payables, receivables, depreciation, appointments, cost accounting, time charges, commissions, inventory, manufacturing control, and even matrix accounting systems!

SUPER PERFORMANCE AT A SUPER PRICE - That SUPER beats the $700 dBASE program may surprise you, but in terms of price vs. performance SUPER has no competitors. Among its features are: production input, data compression, multiple databases on line, transaction posting, file reformating, stored arithmetic files, flexible report formats, hierarchical sort and multi-disk files for up to 131, 068 records. It can select by ranges, sub-strings, and field comparisons. It interfaces to word processors such as WordStar™, SuperSCRIPTSIT™, Model II/16 SCRIPTSIT™, and NEWSCRIPT™. In fact SUPER has so many features that it takes a six-page product description to cover them all. Write or call and we'll send you one.

SUPER is available for TRS-80™ Models I & III under NEWDOS™, LDOS™, and DOSPLUS; for TRS-80 Models II, III and 16 under TRSDOS™; and CP/M™ systems.

Prices:		
	TRS-80 and Osborne versions	$250.00
	Other CP/M versions	$295.00
	Manual (Price applicable to purchase)	$ 25.00

NOW AVAILABLE FOR IBM-PC $250.00

MasterCard and VISA accepted.

OTHER SOFTWARE

- ManageMint™: A PERT/CPM project management system compatible with SUPER. It includes scheduling, resource and financial management modules.
- Sales Planning and Data Extraction System: Improves hit rates while cutting costs.
- Small, economical program packages for accounting, business and office applications as well as utilities.

Write for Catalogue

ISA

Institute For Scientific Analysis, Inc.

SOFTWARE FOR HARD USE ™

Dept. U-1 Institute for Scientific Analysis, Inc.
P.O. Box 7186 Wilmington, DE 19803 (215) 358-3735
Orders only: (800) 441-7680, ext. 500

CDC, 13715 Vanowen St., Van Nuys, CA 91405 (213) 873-6621
Outside of California: (800) 692-5235

Two-part

Step Number	Decimal Code	6809 Instructions
76	3	3
77	57	RTS
78	129	CMPA#
79	254	254
80	38	BNE
81	10	STEP# 92
82	174	LDX 1,U
83	65	PB
84	48	LEAX -1,X
85	31	PB
86	38	BNE
87	252	STEP# 84
88	51	LEAU 2,U
89	66	PB
90	32	BRA
91	219	STEP#55
92	174	LDX ,U
93	196	PB
94	236	LDD 1,U
95	65	PB
96	151	STA$
97	244	244
98	215	STB$
99	246	246
100	10	DEC$
101	244	244
102	38	BNE
103	23	STEP# 127
104	204	LDD#
105	120	120
106	1	1
107	216	EORB$
108	245	245
109	39	BEQ
110	5	STEP# 16
111	187	ADDA$
112	255	255
113	32	32
114	32	BRA
115	4	STEP# 120
116	176	SUBA$
117	255	255
118	32	32
119	64	NEGA
120	183	STA$
121	255	255
122	32	32
123	166	LDA 1,U
124	65	PB
125	221	STD$
126	244	244
127	10	DEC$
128	246	246
129	38	BNE
130	174	STEP# 49
131	204	LDD#
132	120	120
133	1	1
134	216	EORB$

Step Number	Decimal Code	6809 Instructions
135	247	247
136	39	BEQ
137	5	STEP# 143
138	187	ADDA$
139	255	255
140	32	32
141	32	BRA
142	4	STEP# 147
143	176	SUBA$
144	255	255
145	32	32
146	64	NEGA
147	183	STA$
148	255	255
149	32	32
150	166	LDA 2,U
151	66	PB
152	221	STD$
153	246	246
154	32	BRA
155	149	STEP# 49

Program Listing for Two-part Harmony

```
5 REM- TRS-80 COLOR COMPUTER TWO NOTE MU
SIC PROGRAM- LEO CHRISTOPHERSON 1983
10 GOSUB1000: REM- DEFINE STRINGS AND V
ARIABLES
15 GOSUB10000: REM- PACK DATA INTO STRIN
GS
18 REM
19 REM- PLAY PARTS (1), (2), (3), (4), (
1), (2), (3), AND (5)
20 POKE248,A1: POKE249,A2: EXEC MUSIC
21 POKE248,B1: POKE249,B2: EXEC MUSIC
22 POKE248,C1: POKE249,C2: EXEC MUSIC
23 POKE248,D1: POKE249,D2: EXEC MUSIC
24 POKE248,A1: POKE249,A2: EXEC MUSIC
25 POKE248,B1: POKE249,B2: EXEC MUSIC
26 POKE248,C1: POKE249,C2: EXEC MUSIC
27 POKE248,E1: POKE249,E2: EXEC MUSIC
100 STOP
998 REM
999 REM- STRINGS AND VARIABLES
1000 SS$="..............(.156.PERIODS.)
.....
.....
.....
....."
1001 MUSIC=PEEK(VARPTR(SS$)+2)*256+PEEK(
VARPTR(SS$)+3)
1005 PART$(1)="..........(.46.PERIODS.)
....................."
1006 A1=PEEK(VARPTR(PART$(1))+2): A2=PEE
K(VARPTR(PART$(1))+3)
1010 PART$(2)="..........(.46.PERIODS.)
....................."
```

```
1011 B1=PEEK(VARPTR(PART$(2))+2): B2=PEE
K(VARPTR(PART$(2))+3)
1015 PART$(3)="..........(.49.PERIODS.)
.........................."
1016 C1=PEEK(VARPTR(PART$(3))+2): C2=PEE
K(VARPTR(PART$(3))+3)
1020 PART$(4)="..........(.46.PERIODS.)
......................."
1021 D1=PEEK(VARPTR(PART$(4))+2): D2=PEE
K(VARPTR(PART$(4))+3)
1025 PART$(5)="..........(.45.PERIODS.)
....................."
1026 E1=PEEK(VARPTR(PART$(5))+2): E2=PEE
K(VARPTR(PART$(5))+3)
1030 RETURN
1998 REM
1999 REM   DATA FOR SOUND ROUTINE IN LIN
E 1000
2000 DATA 222, 248, 51, 93, 19, 134, 254
, 180, 255, 3, 183, 255
2001 DATA 3, 127, 255, 32, 134, 247, 180
, 255, 1, 183
2002 DATA 255, 1, 134, 247, 180, 255, 3,
 183, 255, 3
2003 DATA 134, 8, 186, 255, 35, 183, 255
, 35, 79, 198
2004 DATA 1, 31, 1, 151, 245, 151, 247,
48, 31, 38
2005 DATA 47, 51, 67, 166, 196, 129, 255
, 38, 17, 134
2006 DATA 247, 180, 255, 35, 183, 255, 3
5, 134, 1, 186
2007 DATA 255, 3, 183, 255, 3, 57, 129,
254, 38, 10
2008 DATA 174, 65, 48, 31, 38, 252, 51,
66, 32, 219
2009 DATA 174, 196, 236, 65, 151, 244, 2
15, 246, 10, 244
2010 DATA 38, 23, 204, 120, 1, 216, 245,
 39, 5, 187
2011 DATA 255, 32, 32, 4, 176, 255, 32,
64, 183, 255
2012 DATA 32, 166, 65, 221, 244, 10, 246
, 38, 174, 204
2013 DATA 120, 1, 216, 247, 39, 5, 187,
255, 32, 32
2014 DATA 4, 176, 255, 32, 64, 183, 255,
 32, 166, 66
2015 DATA 221, 246, 32, 149
2998 REM
2999 REM- DATA FOR MUSIC PART (1)
3000 DATA 16, 63, 75, 16, 71, 85, 16, 76
, 96, 16
3001 DATA 63, 75, 16, 47, 63, 16, 37, 46
, 16, 63
3002 DATA 75, 16, 47, 63, 16, 37, 46, 16
, 30, 36
3003 DATA 16, 37, 46, 254, 60, 16, 37, 4
6, 254, 60
3004 DATA 16, 37, 46, 254, 60, 255
3998 REM
3999 REM- DATA FOR MUSIC PART (2)
4000 DATA 16, 37, 46, 16, 30, 36, 16, 37
, 46, 254
4001 DATA 60, 16, 37, 46, 16, 30, 36, 16
, 37, 46
4002 DATA 254, 60, 16, 37, 46, 16, 30, 3
6, 16, 31
4003 DATA 49, 16, 35, 41, 16, 41, 49, 16
, 50, 84
4004 DATA 16, 63, 101, 254, 60, 255
4998 REM
4999 REM- DATA FOR MUSIC PART (3)
5000 DATA 8, 63, 101, 8, 56, 95, 16, 63,
 101, 16
5001 DATA 71, 85, 16, 63, 71, 16, 50, 62
, 16, 41
5002 DATA 49, 16, 63, 71, 16, 50, 62, 16
, 41, 49
5003 DATA 16, 31, 49, 16, 24, 40, 254, 6
0, 16, 24
5004 DATA 40, 254, 60, 16, 24, 40, 254,
60, 255
```

```
5998 REM
5999 REM- DATA FOR MUSIC PART (4)
6000 DATA 16, 27, 46, 16, 31, 49, 16, 24
, 40, 254
6001 DATA 60, 16, 27, 46, 16, 31, 49, 16
, 24, 40
6002 DATA 254, 60, 16, 27, 46, 16, 31, 4
9, 16, 27
6003 DATA 46, 16, 30, 36, 16, 37, 46, 16
, 47, 63
6004 DATA 16, 63, 75, 254, 60, 255
6998 REM
6999 REM- DATA FOR MUSIC PART (5)
7000 DATA 16, 24, 40, 16, 27, 46, 16, 31
, 49, 254
7001 DATA 60, 16, 24, 40, 16, 27, 46, 16
, 31, 49
7002 DATA 16, 35, 55, 16, 37, 62, 16, 42
, 70, 16
7003 DATA 47, 75, 254, 80, 16, 47, 75, 2
54, 80, 24
7004 DATA 47, 75, 254, 60, 255
9998 REM
9999 REM- PACK SOUND STRING WITH DATA
10000 RESTORE: FORN=MUSIC TO MUSIC+155:R
EADD:POKEN,D:NEXTN
10003 REM
10004 REM- PACK MUSIC PART STRINGS WITH
DATA
10005 A0=A1*256+A2: FORN=A0 TO A0+45: RE
ADD: POKEN,D: NEXTN
10010 B0=B1*256+B2: FORN=B0 TO B0+45: RE
ADD: POKEN,D: NEXTN
10015 C0=C1*256+C2: FORN=C0 TO C0+48: RE
ADD: POKEN,D: NEXTN
10020 D0=D1*256+D2: FORN=D0 TO D0+45: RE
ADD: POKEN,D: NEXTN
10025 E0=E1*256+E2: FORN=E0 TO E0+44: RE
ADD: POKEN,D: NEXTN
10030 RETURN
19998 REM
19999 REM- ROUTINE TO TUNE NOTES
20000 TEST$="....":T1=PEEK(VARPTR(TEST$)
+2): T2=PEEK(VARPTR(TEST$)+3)
20005 TEST=T1*256+T2: POKE248,T1: POKE24
9,T2
20010 POKE TEST,253: POKE TEST+3,255: GO
SUB10000
20015 INPUT A,B: POKE TEST+1,A: POKE TES
T+2,B
20020 POKE MUSIC+132,10: EXEC MUSIC: POK
E MUSIC+132,120: GOTO20015
```

Recovering lost Model II programs

Use DEBUG to get back to BASIC

Model II

Brad Hoza, Puyallup, WA

Model II BASIC programmers: help is here! If you've done much programming in BASIC on the Model II, chances are you've experienced one of the most frustrating things there is: losing your program because the system bombed and you didn't have your program saved.

More than once I have typed SAVE "file" only to have the DOS come back and tell me ?AO Error (File Already Open). This impossible condition is just as impossible to fix in BASIC. Sometimes going to TRSDOS Ready works or, if not, pressing reset always does. Either way, the program is lost.

But, I decided I wasn't going to give up easily. I set out to find out how to restore the program by using the DEBUG Utility (a manual BASIC * for those who are familiar with it on other models).

I use TRSDOS 2.0a and am not sure if the memory addresses used will work in other versions. The only problem with different versions would be in locating the program storage area for BASIC programs. For TRSDOS 2.0a, I found that the program storage area begins at 27714 decimal on a 64K Model II. By the way, all addresses given are decimal unless I state otherwise.

Locating Program Storage

First, from TRSDOS Ready, type CLEAR <enter>, DEBUG ON <enter>, and BASIC <enter>. If you do not see "Ready", type "C" (enter is not needed), and you will be in BASIC. Load a BASIC program and make the first line something like: 1 REM ***Here I am!***. Now get into DEBUG by typing SYSTEM "DEBUG". If you are not used to it, DEBUG's display can be confusing and incomprehensible. Don't worry, you will be concerned with only a very small part of it.

The first line is a peachy little title to remind you that you really are in DEBUG. Look at just the very left-most column in the next line. It's a number, probably 2800. The number is in hexadecimal format (base 16, but that's not too important for us).

The cursor is down several lines, next to a "?". Type "D", this converts the numbers to decimal format (e.g the 2800 would now be displayed as 10240). Next type "M", for memory. DEBUG asks "A=.....", where "A" stands for address. Type 27714. Notice that you don't have to press enter. The number in the upper left should now be 27714.

The display is showing you the contents of the computer's memory, starting at address 27714. The first two blocks show the numeric value and the right-hand side shows the ASCII values. If the program storage area of your BASIC begins at 27714, as mine does, the right-hand side of the display should be familiar. The top line will read something like ".....:***Here " and the second line will say "I am!*** ...". It's the first line of the program that we had in BASIC! If you press the downarrow, you might recognize parts of your program.

If you do not see the first line of the program that you had in BASIC, then 27714 is not the beginning of your program storage area. To find out where it is on your machine, use the up and down arrows until you find the exact location. Once you have the exact value, press the ESCape key. Type "C" and presto! You're back in BASIC and typing LIST should show your program.

There is one complication. The beginning of the program storage area changes depending on how many files you declared when entering BASIC (e.g., BASIC -F:6). Each file that was opened takes up 834 bytes of memory. This results in the start of program memory being moved forward. In order to save your "lost" program you must know how

many files were declared when entering BASIC.

Simulating the Crisis

For practice, go into BASIC and load one of your programs. Now, suppose it is some time in the future and you've lost the program you have worked so hard on. Of course it is all freehand work and is not written down anywhere. And the DOS, or BASIC, or both, have gotten confused and won't work. The only way out is to reset. But you remember this article and pull it out. Just follow these simple steps:

1. You must know the number of files declared when going into BASIC. Let X be this number. If no files were declared and you just used BASIC <enter>, X will be zero.

2. Calculate the start of the program storage area with the formula Start = 27714 + (834 * X). Be sure to change the 27714 to its correct value if your computer acted differently in the discussion above.

3. Press reset. When the system asks for the date, make sure you press the hold key and keep it pressed until you see TRSDOS Ready. This is to turn off any "AUTO" commands and go directly into TRSDOS.

4. Type DEBUG ON <enter>.

5. Type BASIC -F:X <enter>. Make sure that you declare the proper number of files (X). You may or may not get into BASIC and see the friendly "Ready" prompt. The DEBUG monitor may seize control of things after you enter the BASIC -F:X command. If you end up in DEBUG, type "C" and you'll proceed directly into BASIC. You must go to BASIC Ready first, so be sure to type "C" when in DEBUG.

6. When you enter BASIC, the program changes the first two bytes of the program storage area to two zeroes. That says to the computer "No program is here". It is also done whenever you type NEW. Your task is to replace those two bytes with the proper values and thus restore your program. Now that BASIC has been properly initialized you need to enter (or re-enter) DEBUG. Type SYSTEM "DEBUG" <enter>.

7. Type "D" for decimal format.

8. Press "F" for find. DEBUG will ask "S=.....", where S stands for start.

9. Type the value of Start + 4 (refer to step 2). For example, when no files are declared, step two gives Start = 27714. So, in this step we would respond 27718. You do not have to press enter. DEBUG then asks "E=.....", where E stands for end.

10. Type 61440. Again, there is no need to press enter. DEBUG then asks "D=....."

11. Type 00 <enter>.

12. The display changes and the number in the upper left part of the display is some number greater than Start. I will refer to that number as N in step 14.

13. Type M and DEBUG asks "A=.....".

14. Type N+1, that is one greater than the number that is now in the upper left of the display. DEBUG will now show this new number in the upper left part of the display.

15. Press the ESCape key and type X to change to hexadecimal format. Note the number in the upper left part of the display. Write it down exactly as it appears. Think of the number as two halves. The first half is the left two digits and the second half is the right two digits. For example, if the number was 7EA2 the two parts are 7E and A2.

16. Type "D" to change back to decimal format and then type "M". DEBUG asks "A=.....".

17. Type the number you got for Start in step 2. Press the F1 key and the cursor should move to the top left of the left block of numbers. The four digits shown should be "00 00".

18. Now type the number you wrote down in step 15, but, enter the two halves in reverse order. If the number was 7EA2 the "00 00" would now be "A2 7E". It is important that you type the number properly. If you make a mistake, use the backspace key.

19. When you are sure that the number is entered correctly, press the F2 key. The cursor moves back down to the "?".

20. Finally, type "C" and you will be back in BASIC. Type LIST and your program will be there. To SAVE the program use the ASCII format. In some cases, only part of the program would be put to disk if ASCII was not specified.

The procedure described by Mr. Hoza is involved but is also a fine tutorial on the use of DEBUG and gives a key to understanding how Model II BASIC works. We made this same question a Puzzler in our May issue. The first working answer came from Lester J. Dietterick of Berwick, PA. He noted that DEBUG ON enter, DEBUG enter, C, CLS enter would put you back into BASIC from TRSDOS Ready and keep all variables intact. We have not verified that this simple procedure will restore your program under all situations. Mr. Jerry Lippey of Los Angeles, CA sent in a short assembly language routine that will return you to BASIC and keep the Break key operating properly on TRSDOS 2.0 and 2.0a. His code is:

```
LD   HL, 0
LD   A, 3
RST  8
LD   HL, 6069
LD   A, 3
RST  8
JP   2800
```

We hope that one of these three methods will prove useful to you. -- Ed.

ENB

A different type of data base manager

Jim Klaproth, Associate editor

Before we can talk intelligently about ENB, we must define the term *relational data base manager.* Most computerists know what a data base manager does — it organizes and stores data in a structured, usable manner, similar to a card file. Take the simple example of a businessman who needs to maintain a customer file. Each card in the file might contain the customer's name, address, credit rating and discount per sale. Another card file might contain all invoices for a certain time period. Each card might have the customer name, the article purchased, the price, and the date of purchase.

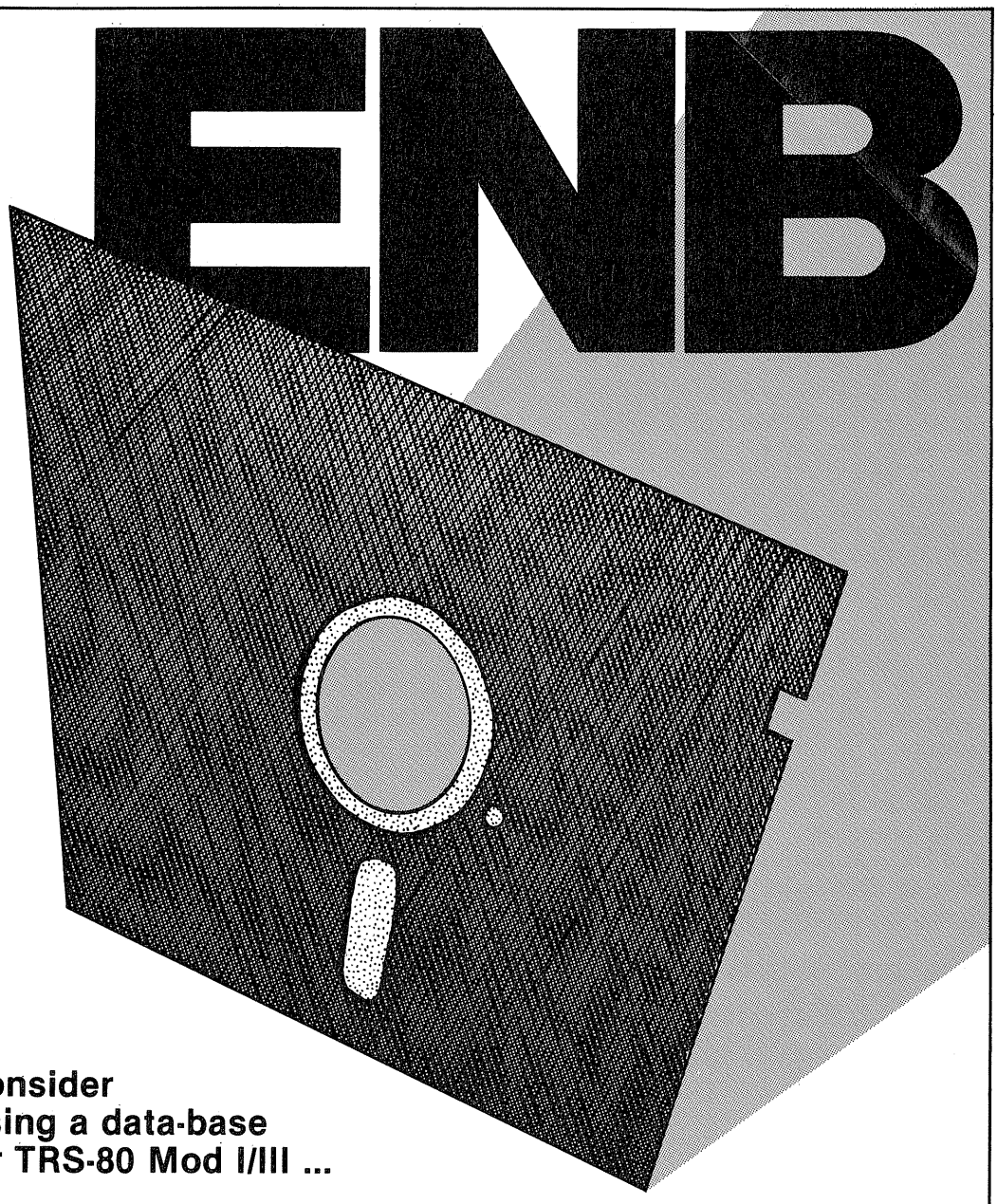
Each data set has a list of attributes like "customer name" and "address" associated with it. In our example of the customer file and invoice file, there exists an attribute common to both files — namely, "customer name." What ENB (or any other relational data base manager) does is recognize this connection and insure that the connection remains consistent throughout the data base. ENB allows the user to store and extract all information relating to any (or all) other attributes in the data base. This allows tremendous power and flexibility in data base management.

For example, using our card files again, say we needed to know the address of a certain customer, but we did not remember his name. We do remember that he purchased a blue widget on Monday of last week. By looking in the invoice file, we could determine all purchasers of blue widgets last week, find the name, and then look up the address in the customer file. Using ENB, we would simply search for a purchaser of a blue widget and list all attributes associated with that purchaser.

ENB is the first relational data base manager to be written for the TRS-80 Models I and III. It was written by a former IBM programmer and, predictably, uses IBM data-structuring techniques. The program uses a machine language core, with a BASIC front-end, to obtain a compromise of speed and easy modification. ENB requires two disk drives on a single-density Model I, but only one drive on a double-density system or a Model III. All current DOS systems may be used, but the excellent disk tutorials work only with TRSDOS systems. This is inconvenient and should be modified to work with all DOS systems. The disk also contains a sample data base and several utility programs.

The sample data base contains data relating to a British school system and it contains some confusing terminology, such as "FORM-ROOM" and "FORM-TEACHER." This is the British equivalent of our "HOME ROOM" and "HOME ROOM TEACHER." Other than the confusing terms, it is an excellent example. The manual guides the neophyte, step-by-step, through the use of each ENB function and the disk tutorials show the student how to type information

ENB



Points to consider when choosing a data-base manager for TRS-80 Mod I/III ...

Data Access Method. ENB has *true* relational access. **Data Independence.** ENB has generalised data structures, no record-length constraints, variable length fields. No data redundancy. **Structure Extension.** Integrated data dictionary permits full editing (add/delete/update/rename) on set of SETS, set of ATTRIBUTES and set of REPORTS. **Data Integrity.** Commit points keep data-base consistant. **Capacity.** 64K distinct data items, spans up to 4 disk drives (or hard disk). **Interface.** Interactive menu-driven entry/display of data. High-level BASIC interface. Scripsit and Visicalc interfaces. **Documentation.** Inbuilt reports automatically document current data-base structure. 125 page manual. Self-running tutorials. **Requirements.** TRS-80 Mod I/III, 48K, at least one disk drive. Works with all DOS. Developed in England by Southern Software.

Scripsit tm Radio Shack, Visicalc tm Visicorp

ALLEN GELDER SOFTWARE
(415) 681-9371
Box 11721, San Francisco, CA 94101

CA add 6%

ENB $140

at each prompt in the program. This is one of the most advanced tutorials we have ever experienced. On the other hand, ENB is the type of program that requires extensive documentation and examples in order to understand its operation.

Perhaps the most difficult task is learning how to set up a new data base. The first step is to identify the objects you are concerned with and then define "sets" to represent them. Each set has to be analyzed to determine all of the connections between sets and to define the attributes and co-attributes (an attribute looked at from the other end). Thus, each time a new attribute is created, a new co-attribute is also formed and may be defined from either end. There are also several pitfalls that have to be overcome in analyzing your data structures. The manual gives some good instruction on how to avoid these obstacles and effectively set up your data base.

Space does not allow a full description of ENB's data structuring or functions. Some of the highlights include numerous formatting of numbered data similar to the PRINT USING function in BASIC, settings for the width of displayed data, and various options on how the data set is handled. These options include proper formatting and sorting of numbers input with special signs (such as the dollar sign), conversion of all input into uppercase, shorthand identification of values (meaning that only enough characters have to be typed in order to uniquely identify the member of the set), closed set option (meaning that no new values may be added or deleted), unnamed members, and invisible members that do not appear to the user but may be useful to the program.

Attributes also have various constraints on how they are handled. There are four different constraints, namely "SINGLE" (each member of the set is allowed only one value), "BELONGS" (members of the value-set belong to the member of the "of-set" e.g., the sales that make up an invoice *belong* to the invoice), "REQUIRED" (every member of the set must have a value assigned to the attribute), and "SECONDARY" (ENB will not prompt you for attributes in the wrong direction). These attribute constraints can be changed, added or deleted any time you find it necessary. Attributes also can be added, deleted, updated, renamed, or listed at any time. This allows a great deal of flexibility and also makes ENB very forgiving to the new user. Most data base managers do not allow changes to the structure after it has been initialized.

Reports are equally flexible in ENB. You can control which columns appear, including multiple attributes and indirect attributes. You control the order of the columns and their widths. You can request column totals and subtotals, and even break the report into pages at logical points. Even the columns may have attributes. These control the information displayed in the column. Of course, these attributes can again be added or deleted to form new report formats at any time. Reports can be displayed on the screen or sent to the printer.

The data base files can span disks and can be expanded up to four logical devices. No problems have been reported when using ENB on a Corvus hard disk system. There are no limitations on number of attributes, other than the practical limit of disk space.

ENB is also compatible with VisiCalc, being able to write files to, or from it. A large section on programming, which rounds out the excellent documentation, allows the advanced user to write his own I/O routines to manipulate the data base. One user we know is doing just that in order to improve on the speed of the program (which is slow compared to a similar program running on his DEC system). In his opinion, ENB is one heck of a fine program for the money and we have to agree. If you need the power that a relational data base manager provides, ENB has no peer at the present time.

ENB Relational Data Base Manager is available from Allen Gelder Software, Box 11721, San Francisco, CA 94101, (415) 681-9371, $140 Models I/III, 32K with disks.

Hashing

Don't let your disk get out of sorts (or how to pour 10 gallons into a 5-gallon hat)

Models I/II/III with disks

Arnold Maddox, Creve Coeur, MO

Once mastered, the TRS-80 random I/O system can be a powerful tool for rapid storage and retrieval of disk data. It has, however, one annoying limitation. The maximum number that can be used as a disk record number (Key) is 335 (Model I). If I managed a store and had parts with identification numbers greater than 335, I couldn't use the part number directly as the key for a random I/O even if I had less than 335 total parts to account for.

Consider a more general example where I have 100 parts that are identified by 9 model numbers, 9 styles, 9 sizes and 9 colors. I would like to store and retrieve data on cost and other information by building a code which could be as large as 9999. For example, part 3827 might be model 3, style 8, size 2 and color 7. How can I store and retrieve data on these parts when the largest random I/O key is 335?

There are several ways to approach this problem. One method would be to dimension a variable in memory to 9999 and have the value of the variable be the actual I/O key. I could then build a "translation" table to convert the actual part number to the key. If part 3827 has a disk key of 22, then A(3827) = 22. A rather simple program will treat this:

```
10 INPUT "ENTER PART NUMBER"; PN
20 KEY = A(PN)
30 GET 1, KEY
40 . . .
```

The user simply enters the part number 3827. This is assigned as the subscript of the A array, and the real key is found and used for the disk GET command. An array of sufficient size would be required to cover all possible part numbers (i.e., 9999). This would be reasonably fast but has two drawbacks. First, a large part of the A array will be empty. In the example, only 100 out of the 9999 would be used. Secondly, the user will have to either build a routine to add and subtract part numbers and their keys, or change the program each time a part is added or deleted from the stock list.

Another approach would be to dimension an array to the actual number of parts with the value of the array being the part number. The program would then loop through the array and compare each array value with the entered part number. In this case part number 3827 with a key of 22 would resemble A(22) = 3827. The following short program would implement this.

```
10 INPUT "ENTER PART NUMBER"; PN
20 FOR I = 1 to 100
30 IF A(I) = PN then 70
40 NEXT I
50 PRINT "NO SUCH PART NUMBER"
60 GOTO 10
70 GET 1, PN
80 . . .
```

This approach has the advantage of a smaller array but, on the average, the program will have to loop through half of the array each time a search is needed. This approach would also have the problems of updating the array as needed.

The approach proposed here uses a technique called "HASHING." No arrays are needed, updating is

virtually automatic, and the number of attempts required to find a part can be much less than the number of items in the file. It has the disadvantage that the disk file must be larger than the number of parts and this may determine the speed of a search. Several hashing algorithms are available but the simplest and most obvious is called the "Remainder Probe."

The steps are outlined below.

1. Divide the part number by the selected size of the file (FS). For this example, a file size of 2.5 times the number of parts seems about right. (A further discussion on file size is at the end of the article.) Looking at the above example: 3827/250 = 15.308. This quotient (Q) of 15 and a remainder (R) of 77. If the quotient is zero, set it to 1.

2. Make a "probe" using the GET command with a key equal to the remainder, i.e., GET 1, R where R = 77.

3. Two things can happen:

A. There is something in the record (this is called a collision) which can be compared with the part number. If it compares favorably, print out the price and other information. If it is not the right part number, "re-hash" by going to step 4.

B. There is nothing there. This is a good place to stick a new part if it needs to be added.

4. Re-hash by computing a new Q and R using (R + Q)/FS or (77 + 15)/250.

This gives a quotient of 0 and a remainder of 92. If the quotient is zero, set it to 1. Now, return to step 2 with this remainder as the key and make another probe. Note that the key can never exceed the file size selected. Even if the part number was 9999, Q = 39 and R = 249. 100 objects with part numbers up to 9999 can be stored in 250 records. Notice that the part number could even be 99999 without any change in the method.

The program in Listing 1 implements this algorithm. Lines 50 - 180 contain a simple driver to demonstrate the subroutine. Lines 5010, 5080 and 5110 were added so the user could follow the progress of the hashing.

In keeping with sound file security, the file is only open during actual reading and writing. Incidently, I think it is good practice to zero out every random file before it is first used. Listing 2 zeroes the file for this example.

To determine the efficiency of the method, one hundred random part numbers (max = 9999) were selected and used to store and retrieve records using the hashing subroutine. The results for three file sizes are shown in the table below.

Table 1

Number of Probes

File Size	1	2	3	4	5	6	7
150	64	21	8	3	2	1	1
200	74	18	5	2	1	0	0
250	77	15	5	2	1	0	0

This table shows that 77% of the time, the record was found on the first probe with a file size of 250; 92% were found by the second probe. As expected, smaller file sizes produced more unsuccessful probes.

Alphanumeric Hashing

In many applications it would be desirable to input the name of a client or club member and use that name to access that person's record:

```
10 INPUT "ENTER PERSON'S NAME"; PN$
20 GET 1, PN$
30 . . .
```

Unfortunately, the TRS-80 does not permit alphabetic or special characters in the key of the GET command. The above code will produce the error message "TYPE MISMATCH." The most common way around this problem would be as shown in Listing 3 (for 100 names). On the average, one-half of the names would have to be compared for each search. A way is needed to convert the name to a numeric value so we can use the hashing subroutine shown last month. The first step is to use the ASC command to convert each letter into its ASCII numeric equivalent. The most obvious way to build a hash number is to use the ASCII numbers directly. The number of letters that can be packed into various numbers is summarized in the table below.

Table 2

Type	Max. Size	Number of Letters
Integer	32768	2½
Single Precision	6 digits	3
Double Precision	16 digits	8

If more letters are placed in the hash number, fewer unsuccessful collisions can be expected. This indicates that double precision is the best. The use of additional letters considerably outweighs the loss of speed produced with double precision arithmetic. The name, PUBLIC, JOHN Q., would translate to: PN # = 8085667673774432.

Listing 4 presents a program to demonstrate this approach. It was necessary to modify the hashing subroutine from Listing 1 to consider both double precision and the string comparison. Significant conceptual changes are in line 115, where it was necessary to make all strings the same length, and lines 5002 to 5008, where the string is converted to a double precision number. Incidentally, since numbers can also be input as strings, this subroutine will do everything the program in the first article will do.

The reader has probably deduced by now that the names will not be in alphabetic order in the file. If the random file is sorted, hashing will not find the right records. The solution to this is outlined below:

1. Open and properly field the file.
2. Find the last record in the file (LOF).
3. Loop through the entire file incrementing the key from 1 to LOF.
4. Build an array composed of the name and key: A$ (KEY) = PN$ + STR $ (KEY).
5. Sort this array using any available sorting method.
6. Loop again through this sorted array, picking off the key. KEY = VAL (RIGHT $ (A$(KEY), 5)).

7. With this key, GET the record for print and/or review.

No hashing is needed since the key will be the actual number between 1 and 335.

Sub-Record Hashing

Another serious limitation and subsequent loss of flexibility with Random DISC I/O is produced by the fact that every record is 255 bytes long whether you need it or not. Even if you used the hashing methods already discussed, you would still be limited to the 335 records of 255 bytes each. The TRSDOS Manual (page 7-71) provides an excellent discussion and examples of how to conduct I/O operations with subrecords.

To aid in the understanding of how hashing and subrecords work, we will use the example below:

Table 3

Data Element	Number of Characters
Name	20
Address	15
City	10
ZIP	5
	50

(For demonstration purposes, all but the name will be called "STUFF.")

Dividing the maximum number of characters in a record by the above total will produce the maximum number of subrecords in a physical record (255/50 = 5.1 or 5 subrecords per physical record). The maximum number of subrecords in the file will be 335 * 5 or 1650. That certainly gets around the 335 record limit, but we still need to hash larger part numbers and alphanumeric data to fit within the limit of 1650. It is therefore necessary to merge the subrecord and hashing techniques. The program in Listing 5 accomplishes this.

The first major change needed to the program in Listing 4 is to dimension all of the variables needed for the subrecord structure (line 40). Secondly, it is necessary to require inputs of record size and number of records to be used in the file under consideration (lines 20 to 35). Lines 5022 to 5026 perform the FIELDing. Lines 5082 and 5084 calculate the physical and subrecord indices after the alphanumeric string has been converted to a double precision number. Finally, it is necessary to add the subscript SR% to all parameters in the FIELD (i.e., PN$ & P$).

We now have a method to (almost) directly access disk records using strings and large numeric values. The only arrays that are necessary are associated with fielding subrecords. No program changes are necessary to add new names or part numbers to the files. We do not need the space and run time to load translation tables required by alternate methods.

Listing 1

```
1 CLS
50 INPUT"ENTER FILE SIZE";FS
100 INPUT"ENTER CODE (1 FOR STORE, 2 FOR
 SEARCH, 3 FOR DELETE)";C
110 INPUT"ENTER PART NUMBER";PN
120 IFC=1THENINPUT"ENTER PRICE AND OTHER
 STUFF";P
130 GOSUB5000
140 IF RF>0 THEN170 '
WAS THERE ANYTHING THERE
150 PRINT"PART NOT IN FILE"
160 GOTO100
170 IFC=2 THEN PRINT "PRICE ON PART NUMB
ER ";PN;" IS $";P
180 GOTO100
5000 '*** HASH IT ***
5010 RF=0'
SET RETURN FLAG
5020 OPEN "R",1,"TEST"'
 OPEN IT
5030 FIELD 1,2 AS PN$, 6 AS P$'
FIELD IT LIKE YOU WANT
5040 K=PN'
SAVE THE PN
5050 Q=INT(K/FS)'
CALCULATE Q
5060 R=INT(K-Q*FS)'
CALCULATE R
5070 IFQ=0 THEN Q=1'
ADJUST Q IF NEEDED
5080 PRINT"K=";K;"Q=";Q;"R=";R;"COUNT=";
CT;:CT=CT+1
5090 GET 1,R
5100 LT=CVI(PN$)'
DECODE PART NUMBER
5110 PRINT"LT=";LT
5120 IF C > 1 THEN 5190'
FOR SEARCH AND DELETE
5130 IF LT <> 0 THEN 5280'
NOT EMPTY FOR STORE
5140 LSET PN$ = MKI$(PN)'
     FOUND AN EMPTY ONE
5150 LSET P$  = MKS$( P)'
       GET READY TO STORE
5160 PUT 1,R'
          STORE IT
5170 CLOSE
5180 RETURN
```

```
5190 IF LT<>0 THEN5230 '
IT SHOULD BE THERE
5200 RF=-1'
    BUT IT'S GONE.
5210 CLOSE'
    SEND A MESSAGE BACK
5220 RETURN
5230 IF LT<>PN THEN5280
5240 IF C=3 THEN PN=0: GOTO 5140'
BLANK IT OUT
5250 P=CVS(P$)'
DECODE THE PRICE
5260 CLOSE
5270 RETURN
5280 K=Q+R'
REHASH
5290 GOTO5050
```

Listing 2

```
1 CLS
5 CLEAR 1000
10 INPUT"ENTER FILE SIZE";FS
20 OPEN"R",1,"TEST"
30 FIELD1,255 AS PN$
50 FOR R=1 TO FS
60 PRINT@65,R
70 A$=STRING$(255," ")
80 LSET PN$ = A$
90 PUT1,R
100 NEXT R
110 CLOSE
120 END
```

Listing 3

```
10 DIM NM$(100)'                    ALLOW FOR
100 NAMES
20 INPUT "ENTER PERSON'S NAME";PN$
30 FOR PN = 1 TO 100'               LOOK THROU
GH ALL NAMES
40 IF NM$(I) = PN$ THEN 80'         FIND THE R
IGHT ONE?
50 NEXT PN'                           GUESS NO
T!
60 PRINT "NAME NOT IN FILE"
70 GOTO 20'                         TRY AGAIN
80 GET 1, PN'                       GO GET REC
ORD
90 ........'                        REST OF PR
OGRAM
```

Listing 4

```
1 CLS
10 CLEAR1000
50 INPUT"ENTER FILE SIZE";FS
100 INPUT"ENTER CODE (1 FOR STORE, 2 FOR
 SEARCH, 3 FOR DELETE)";C
110 INPUT"ENTER NAME";NM$
115 NM$=NM$+STRING$(20-LEN(NM$)," ")'
MAKE IT 20 LONG
120 IFC=1 THEN INPUT"ENTER OTHER STUFF";
STUFF$
130 GOSUB5000
140 IF RF>0 THEN170
 WAS THERE ANYTHING THERE
150 PRINT"NAME NOT IN FILE"
160 GOTO100
170 IFC=2 THEN PRINT "STUFF FOR ";NM$;"
IS ";STUFF$
180 GOTO100
5000 '*** HASH IT ***
5002 K#=ASC(LEFT$(NM$,1))'
CONVERT
5004 FOR I=2 TO 8'
  ALPHA
5006 K#=K#*100+ASC(MID$(NM$,I,1))'
     TO
5008 NEXT I'
      DOUBLE PREC.
5010 RF=1'
SET RETURN FLAG
5020 OPEN "R",1,"TEST"'
 OPEN IT
5030 FIELD 1,20 AS PN$, 6 AS P$'
FIELD IT LIKE YOU WANT
5050 Q#=INT(K#/FS)'
CALCULATE Q
5060 R#=INT(K#-Q#*FS)'
CALCULATE R
5070 IFQ#=0 THEN Q#=1'
ADJUST Q IF NEEDED
5080 R% = R#
5090 GET 1,R%
5100 LT = 1'
SET THE EMPTY FLAG
5110 IFPN$=STRING$(20," ")THEN LT=0'
 IF EMPTY FLAG =0
5120 IF C > 1 THEN 5190'
FOR SEARCH AND DELETE
5130 IF LT <> 0 THEN 5280'
NOT EMPTY FOR STORE
```

```
5140 LSET PN$ = NM$'
FOUND AN EMPTY ONE
5150 LSET P$  = STUFF$'
   GET READY TO STORE
5160 PUT 1,R%'
     STORE IT
5170 CLOSE
5180 RETURN
5190 IF LT<>0 THEN5230 '
IT SHOULD BE THERE
5200 RF=-1'
   BUT IT'S GONE.
5210 CLOSE'
    SEND A MESSAGE BACK
5220 RETURN
5230 IF PN$ <> NM$ THEN 5280'
IS IT THE ONE?
5240 IFC=3THENNM$=STRING$(20," "): GOTO
5140'        BLANK IT OUT
5250 STUFF$ = P$'
GET STUFF FROM BUFFER
5260 CLOSE
5270 RETURN
5280 K# =Q# + R#'
      REHASH
5290 GOTO5050
```

Listing 5

```
1 CLS
10 CLEAR1000
20 INPUT"ENTER CHARACTERS IN YOUR DATA";
NC
25 NF=INT(255/NC)
30 INPUT"ENTER MAX PHYSICAL RECORDS";FS
35 FS=FS* NF
40 DIM PN$(NF), P$(NF)
100 INPUT"ENTER CODE (1 FOR STORE, 2 FOR
 SEARCH, 3 FOR DELETE)";C
110 INPUT"ENTER NAME";NM$
115 NM$=NM$+STRING$(20-LEN(NM$)," ")'
MAKE IT 20 LONG
120 IFC=1 THEN INPUT"ENTER OTHER STUFF";
STUFF$
130 GOSUB5000
140 IF RF>0 THEN170  '
 WAS THERE ANYTHING THERE
150 PRINT"NAME NOT IN FILE"
160 GOTO100
170 IFC=2 THEN PRINT "STUFF FOR ";NM$;"
IS ";STUFF$
180 GOTO100
5000 '*** HASH IT ***
5002 K#=ASC(LEFT$(NM$,1))'
CONVERT
5004 FOR I=2 TO 8'
  ALPHA
5006 K#=K#*100+ASC(MID$(NM$,I,1))'
    TO
5008 NEXT I'
      DOUBLE PREC.
5010 RF=1'
SET RETURN FLAG
5020 OPEN "R",1,"TEST"
5022 FOR I% = 0 TO NF-1
5024 FIELD 1, (I%*NF) AS SH$, 20  AS PN$
(I%), 30 AS P$(I%)
5026 NEXT I%
5050 Q#=INT(K#/FS)'
CALCULATE Q
5060 R#=INT(K#-Q#*FS)'
CALCULATE R
5070 IFQ#=0 THEN Q#=1'
ADJUST Q IF NEEDED
5080 R% = R#
5082 PR% = INT((R%-1)/NF)+1'            P
HYSICAL RECORD
5084 SR% = R% - NF*(PR%-1)-1'
 SUB-RECORD
5090 GET 1,PR%
5100 LT = 1'
SET THE EMPTY FLAG
5110 IFPN$(SR%)=STRING$(20," ")THEN LT=0
'      IF EMPTY FLAG =0
5120 IF C > 1 THEN 5190'
FOR SEARCH AND DELETE
5130 IF LT <> 0 THEN 5280'
NOT EMPTY FOR STORE
5140 LSET PN$(SR%) = NM$'
     FOUND AN EMPTY ONE
5150 LSET P$(SR%)  = STUFF$'
        GET READY TO STORE
5160 PUT 1,PR%'
      STORE IT
5170 CLOSE
5180 RETURN
5190 IF LT<>0 THEN5230 '
IT SHOULD BE THERE
5200 RF=-1'
   BUT IT'S GONE.
5210 CLOSE'
    SEND A MESSAGE BACK
5220 RETURN
5230 IF PN$(SR%) <> NM$ THEN 5280'
     IS IT THE ONE?
5240 IFC=3THENNM$=STRING$(20," "): GOTO
5140'        BLANK IT OUT
5250 STUFF$ = P$(SR%)'
     GET STUFF FROM BUFFER
5260 CLOSE
5270 RETURN
5280 K# =Q# + R#'
      REHASH
5290 GOTO5050
```

Put 64K CP/M® 2.2 in your TRS-80 Model III and tap into 2,000 business programs.

Now you can run programs such as WordStar, dBASE II, SuperCalc, MailMerge and virtually thousands of other CP/M-based programs on your TRS-80 Model III.

CP/M 2.2 is the industry standard operating system that gives you access right now to over 2,000 off-the-shelf business programs.

Our plug-in Shuffleboard III comes with 16K of RAM, giving your Model III the power of full 64K CP/M 2.2 without interference of the ROM or video memory. In fact, the Shuffleboard will appear transparent in the TRS-80 mode and will not interfere with any DOS operation.

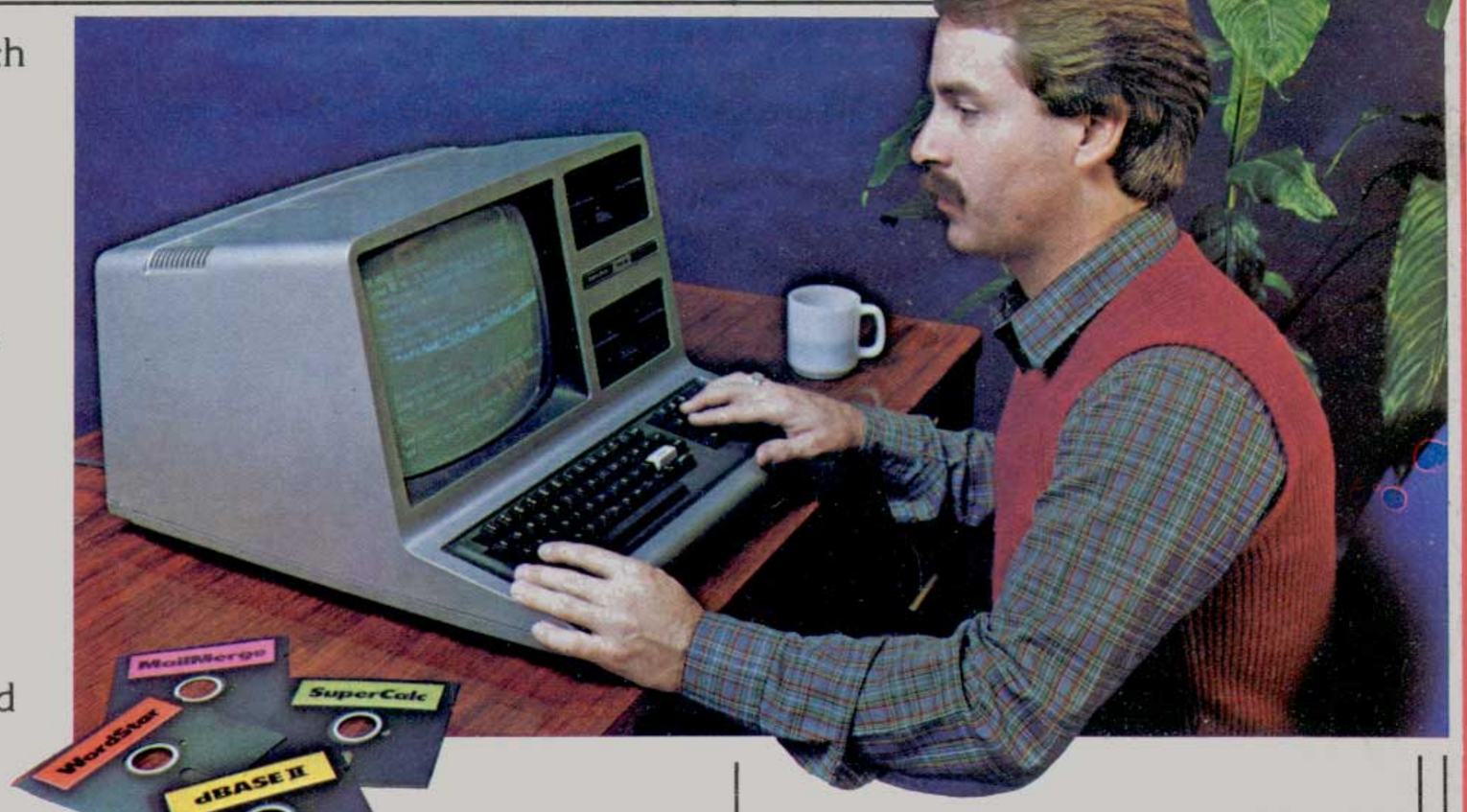
READ and WRITE Osborne, Xerox and IBM personal computer software plus many more popular formats.

Unfortunately, there is no standardized CP/M format for 5¼" diskettes. But we have developed a way to READ/WRITE and RUN standard programs under the following single-sided formats: Osborne 1 S/D, Xerox 820 S/D, IBM PC* D/D for CP/M 86 only, Superbrain D/D, Kapro II D/D, HP 125 D/D and TeleVideo D/D.

*Will Read and Write Only.

Easy plug-in installation.

It's so simple. The Shuffleboard III plugs into two existing sockets inside your Model III. There are no permanent modifications, no cut traces and no soldering. You'll be up and running in minutes.

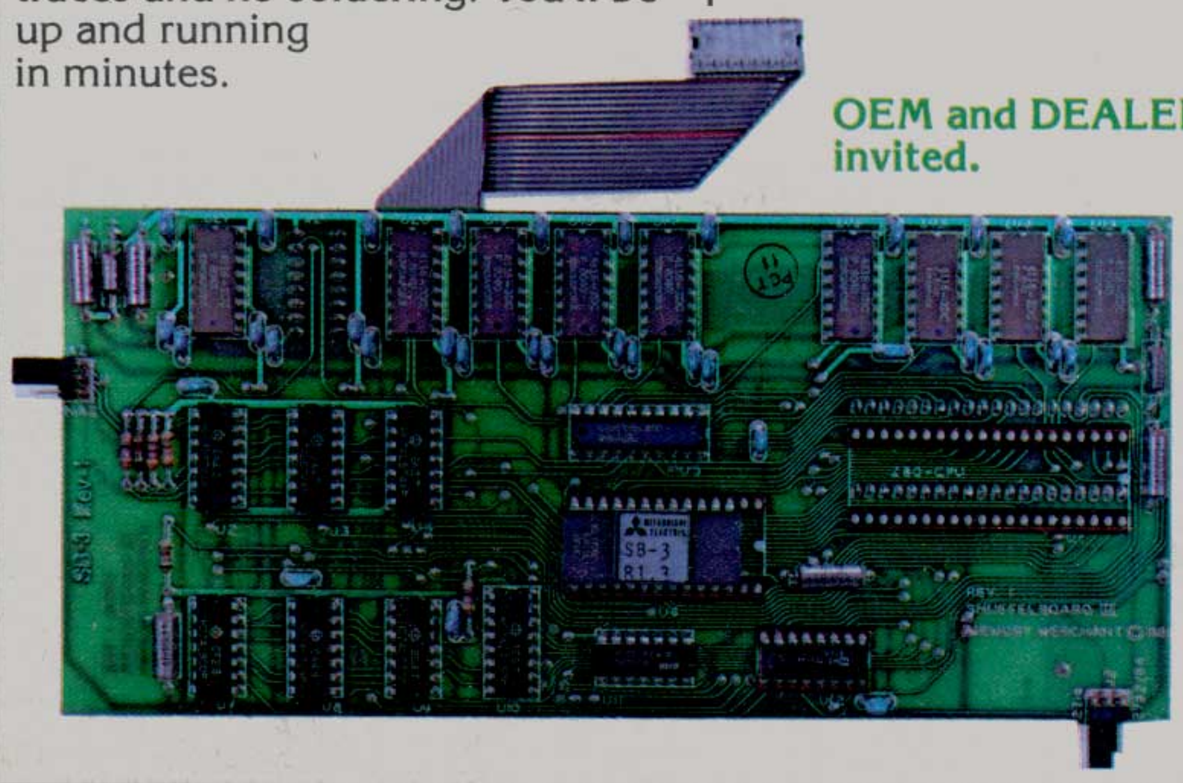


New Products.

80 × 24 VIDEO BOARD: Features dual intensity screen, programmable cursor control for block, underline & blink rate, on-board bell with audible keyclick, battery-operated real time calendar/clock, full ASCII character set plus 256 special character graphics, dual RS-232 outputs and composite video output.

FLOPPY DISK CONTROLLER: Now you can access 5¼" and 8" floppy disk drives in any combination up to 4 drives of S/D density, S/D sided. Tap into a wealth of CP/M software which comes on 8" IBM 3740 format or Pickles & Trout CP/M for the Model II.

SOFTWARE: Additional CP/M software programs are available. Call or write for details.



OEM and DEALER inquiries invited.

Introductory price of $299.

The Shuffleboard III comes fully burned-in and tested complete with 64K CP/M 2.2 and MBASIC 80 interpreter, plus software manuals and a first class user's manual — with a 1-year limited warranty and 15-day no-risk free trial — for only $299.

See the Shuffleboard III at your dealer's now.

Once you see what the Shuffleboard can do for your Model III you'll want one at once. If your dealer does not yet stock the Shuffleboard have him give us a call. Or send check, money order, VISA or MASTERCARD number (sorry, no COD's) plus $5 shipping per board ($17 outside the USA & Canada)* directly to the address below. Cal. residents please add sales tax. Credit card purchases can be phoned in directly and we'll ship from stock.

(415) 483-1008

*Air mail shipments to Canada & all other countries.

Memory Merchant™

14666 Doolittle Drive San Leandro, CA 94577
(415) 483-1008

WordStar & MailMerge are trademarks of MicroPro.
SuperCalc is a trademark of SORCIM.

dBASE II is a trademark of Ashton-Tate.
CP/M is a trademark of Digital Research.

TRS-80 is a trademark of Tandy Corporation.
IBM is a trademark of IBM Corporation.

BASIC bits

Speeding up your string handling

Models I/III

Thomas L. Quindry, Burke, VA

Feedback from readers indicates that perhaps the most interest in this column is generated when string handling techniques are discussed. String handling probably accounts for the slowness of most programs and any way to speed up this process is most helpful.

Tom Ferris of Rowland Heights, CA, gave me the following tip when using Disk BASIC. To avoid delays from going into the garbage handling routine during string manipulation, use the MID$= function on the lefthand side of your BASIC statement to exchange string values. For example, let's assume A$= "BASIC Bits" and you want to redefine it to A$= "Quindry." Use the following: MID$ (A$,1,10)= "Quindry ". Note that "Quindry " is ten characters long (the name, plus three spaces). The string you are redefining must be the same length as the one it replaces using this tip.

Here is an explanation. If you go back to my second BASIC Bits column, November, 1982, you will see a short listing which illustrates what happens when a string is redefined. String space fills up from higher memory to lower memory. When a string is redefined, the old string is not immediately eliminated. It remains, at least temporarily, in memory. The new string definition is just placed lower in memory within the reserved string area. When the reserved string space is filled up, the active string values are reshuffled and written over the old values. This is what is called garbage collection. Tom pointed out that a different process takes place when using MID$=.

This statement lets you replace any part of a string with a specified new string. Remember that the length of the resultant string is the same as the original string. Realizing this fact, the BASIC interpreter simply replaces the new partial string value within the original string area without relocating it in memory. Since your resultant string is always the same length as the one it replaces, there is no need to just tack it on to the bottom of the used string area. The original location of the string is good enough, so the hassle of potential string garbage collection is eliminated.

The syntax for MID$= is as follows for a string called A$:MID$(A$,X,Y)="xxxxx" where X is the position of the first character of the string you want to replace and Y is the length of the string you want to replace. The designation "xxxxx" represents the number of characters to be replaced and is equal to Y-X+1 in length. If you set X=1 and Y equal to the full length of the string, you can redefine the entire string and avoid garbage collection. To verify this fact, try the following:

```
10 CLS:INPUT"ANSWER QUESTION WITH <BASIC
 BITS> ";A$
20 PRINT A$:GOSUB 100
30 MID$(A$,1,10)="QUINDRY    ":PRINT A$
40 GOSUB 100:PRINT"USING MID$=GIVES NO C
HANGE."
50 A$="BASIC BITS":PRINT A$
60 GOSUB 100:PRINT"NOTICE THAT THE ADDRE
SS CHANGED."
70 END
90 REM SUBROUTINE FINDS LSB AND MSB OF S
TRING ADDRESS
100 PRINT PEEK(VARPTR(A$)+1),PEEK(VARPTR
(A$)+2)
110 RETURN
```

The values printed out by line 100 are used to compute the address of the string.

One restriction is that the original string must be located within the reserved string space or higher, i.e., protected memory. (I'll explain this next month.) The original string cannot be named from within the program. If you changed line 10 to A$= "BASIC Bits" the string variable would be first relocated to string memory and you would see a change in the address location printed by line 40.

This brings up an important tip given by Vince Bly of Alexandria, VA. Vince's tip goes hand-in-hand with the MID$= tip. He points out that once a string has been defined, you can alter the string pointers and even the defined length of the string to give you a different string value located somewhere in memory. This can be illustrated by the following simple program:

```
10 A$="" 'Define dummy string
20 POKE VARPTR(A$),27 'Define new string length
30 POKE  VARPTR(A$)+1,17  'Define  new  LSB  of
address
```

40 POKE VARPTR(A$) +2,1 'Define new MSB of address
50 PRINT A$ 'Surprise!

What we have done is redefine the length of the string, A$, and have pointed it to a new location — this time to a location in ROM. You could have just as easily pointed to any other location in memory including the screen area.

This brings up another interesting concept which Vince pointed out to me. Suppose you don't have Disk BASIC and need a LINEINPUT function? If you define your string to an area of screen memory, anything that is written in that area of the screen temporarily becomes part of that string. You can redefine that string to a more permanent string name which will reside within the reserved string space storage area in high memory. To illustrate this, key in the following program:

```
10 REM EXAMPLE OF CUSTOM INPUT STATEMENT
20 REM CONCEPT BY VINCE BLY
30 CLEAR100:DEFINT C:CLS:PRINT:PRINT:PRI
NT:V=0:SC$="":V=VARPTR(SC$)
40 PRINT"ENTER YOUR NAME : ";:GOSUB 1000
0
50 N$=S$ 'SAVE NAME IN N$
60 PRINT"ENTER YOUR AGE : ";:GOSUB 10000
 'TRY --- 80 YEARS YOUNG
70 IF VAL(S$)=0 THEN 60 ELSE A$=S$:PRINT
:PRINT"HELLO";N$;"!" 'IS THIS A NUMBER?
80 PRINT"ARE YOU REALLY ";A$;"? ";:GOSUB
 10000
90 IF (LEFT$(S$,1)="N")OR(LEFT$(S$,1)="n
") THEN 60
100 IF(LEFT$(S$,1)<>"Y")AND(LEFT$(S$,1)<
>"y") THEN 80
110 PRINT"Good!"
9999 END
10000 CR=PEEK(16416)+256*PEEK(16417):CN=
CR 'Cursor location
10010 POKEV+1,PEEK(16416):POKEV+2,PEEK(1
6417) 'String location
10020 REM Pokes in line 10030 give blink
ing cursor block
10030 POKECN,176:Y$=INKEY$:POKECN,32
10040 IF Y$="" THEN 10030 ELSE IF ASC(Y$
)=13 THEN 10070
10050 PRINTY$;:CN=PEEK(16416)+256*PEEK(1
6417):C=CN-CR
10060 IF C<0 THEN Y$=" ":GOTO 10050 ELSE
 IF C>24 THEN 10070 ELSE 10030
10070 POKEV,C:S$=SC$:PRINT:RETURN 'set s
tring length
10080 'it is important to redesignate st
ring to s$
```

One interesting thing about the above program that I can't explain and haven't had the time, nor inclination, to explore, is that if DEFINT C is omitted in line 30, the cursor will blink faster. Also, if you POKE CN+0 ... at both places in line 10030, you also get a faster cursor. Do both of these and you really get a fast blink? If someone can explain this phenomenon, I'd be glad to tell other readers.

To some extent, these may seem like two nice, useful (but not too powerful) tips. But, they are very powerful. Using these two tips as a basis, I was able to write a program which was capable of handling over 2000 string values and never went into the garbage collection routine under normal use. I had a reserved string space of 26000 bytes and used a LINEINPUT routine similar to the above, rather than the INPUT or LINEINPUT function available in Disk BASIC. I then redefined inputs to already-allocated strings using the MID$= tip. No string variable ever had a chance to be redefined normally, which would eat up string space. With a string area that large, going through garbage collection would take 11 to 12 minutes each time on a Model III.

Next month, more on these two powerful tips. I'll give you two simple screen dump routines to a lineprinter (courtesy of Vince) and also a short program which will be especially useful to single drive disk users.

Remember to send your requests for future column topics, questions and tips to me, care of *Basic Computing*, 3838 South Warner Street, Tacoma, WA 98409. Send a self-addressed stamped envelope and I'll try to give you a personal, handwritten reply if the answer is not too long and involved. Problems of general interest may be included in future BASIC Bits.

In the chips

Wall-to-wall bytes

Models I/III

Spencer Hall

Good morning class! It's good to see everyone here. You are all probably anxious to write some machine language. We'll get to it in a moment, but first a bit of unfinished business. Last time, you were shown the list of reserved words in ROM, which you could see with our BASIC monitor. They are even more visible lying there from address 1650H (5712 decimal) on, when viewed with DEBUG. You were asked to figure out why the first letters of these words were missing.

The answer is that they are not missing. They are simply disguised. The bytes in this region are, of course, the ASCII values for capital letters. They are, as you will remember from our earlier columns, actually written as eight-digit binary numbers. The ASCII value for the first letter of each reserved word has been modified by "turning on" bit number seven, the one representing 128. You can prove this to yourself by subtracting 128 from the ASCII for any "missing" first letter. The result will be the ASCII code for the missing letter.

Now to machine language. We're going to fill the screen with the "and" sign, otherwise known as the ampersand, and written &. Sorry it's nothing fancier, but there's a lot to learn about machine language from this simple exercise.

Let's ask ourselves what must be done. The screen is "mapped" to addresses 15360 through 16383, as we saw last month. Each of these addresses contains the ASCII byte for what appears at the corresponding location on the screen. The ASCII code for the ampersand is decimal 38. All we need to do, then, is to put this byte in each address of the screen memory.

Let's do this first in BASIC. The code in Listing 1 is one way. Its logic is identical to that which we are going to use in our machine language version. Please type in this little program, run it and save it. No need, of course, to type to remark statements. They are only for use in this discussion. There are other ways to fill the screen in BASIC. There are even more ways in machine language. We've chosen this way because it illustrates some things you need to learn first.

Run this program and time it. You will observe that the screen takes almost exactly nineteen seconds to fill. Our machine language program will produce the same result in .024 seconds. It could fill the screen 792 times while the BASIC program is filling it once!

One-by-one, we'll convert each logical step in our BASIC program to its machine language equivalent.

Make SA Point to the First Screen Address

There is a memory location inside the Z-80 known as the HL register. (This is not a RAM chip, remember, but the Z-80 chip itself.) It looks and acts like a sixteen-bit register, but it is actually a pair of eight-bit registers, H and L, containing (you guessed it) the high and low bytes of a binary integer. HL will be your most-often used register pair because the Z-80 can do the most things with it. It is most often used as a "pointer," to contain a RAM address which the programmer plans to use in some way. You guessed it again. We're going to make it point to 15360, exactly as variable SA did in our BASIC version. Here are three versions of this statement:

English: Load register HL with 15360.
Object Code: 21 00 3C
Source Code: LD HL,3C00H

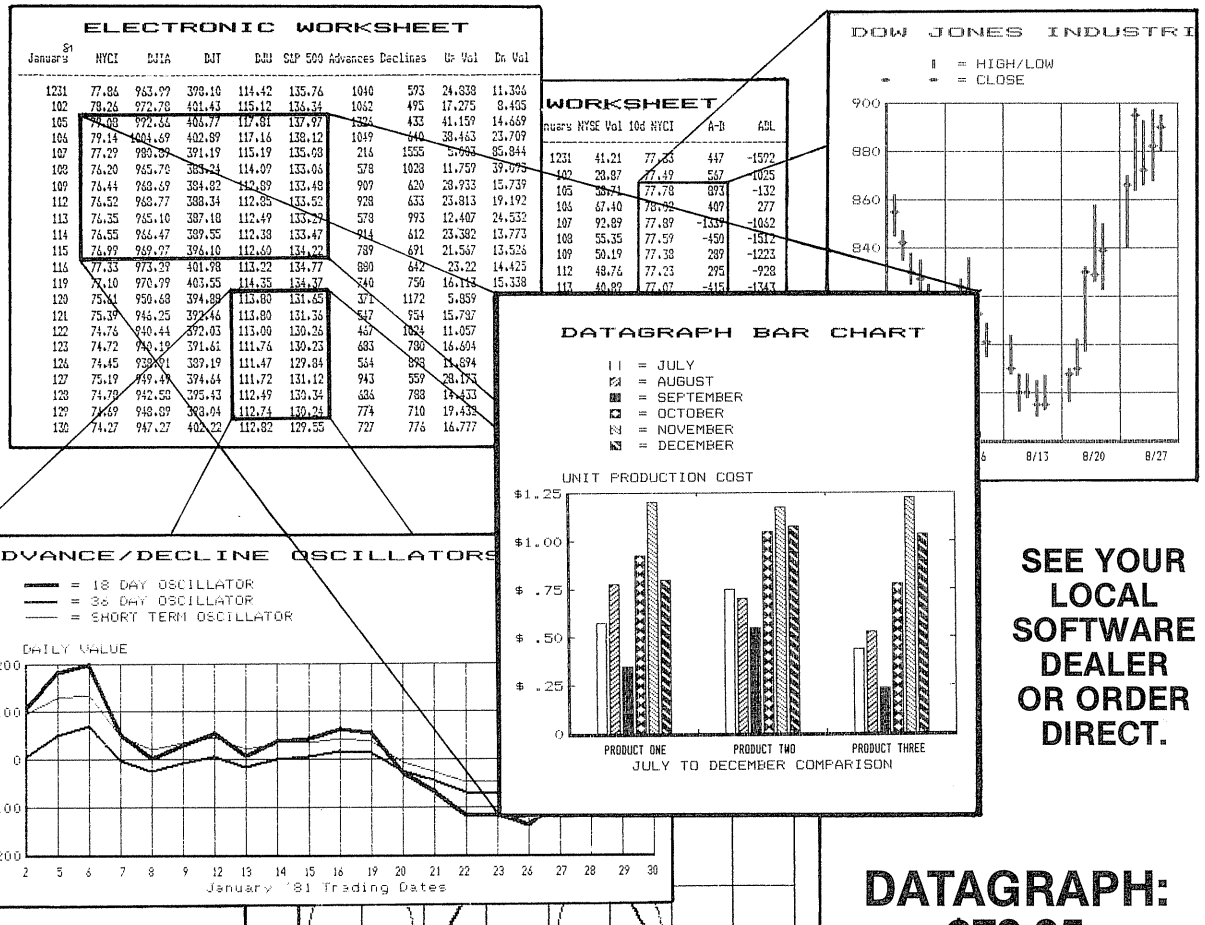
The object code consists of three bytes which will "reside" one after the other, in RAM at the starting address of our machine language program. You can recognize the second and third bytes as the LSB/MSB (reversed order) version of 3C00, which is hexadecimal for 15360. The hexadecimal byte 21 is machine code meaning "put the next two bytes in the HL register." How, in heaven's name, are you going to remember *that*? You aren't going to! This month is the only time in your life you're going to write machine language without an editor/assembler (TRS-80 versions are called EDTASM). EDTASM understands the language of the source code version and translates it into object code. This process is called *assembling*. Actually, EDTASM does a great deal more for you — so much more that you would have to be crazy to attempt machine language programming without it. (Please! No smart remarks!)

Whenever bytes are being placed in a Z-80 register, or in RAM, source code uses the mnemonic, LD (for LOAD),

DATAGRAPH*

T.M. 3RD GENERATION PRINTER GRAPHICS PROGRAM

TRANSFORM YOUR VISICALC™ FILES INTO HIGH-RESOLUTION CUSTOM GRAPHS ON YOUR TRS-80™ COMPUTER AND GRAPHICS PRINTER.



SEE YOUR LOCAL SOFTWARE DEALER OR ORDER DIRECT.

DATAGRAPH: $79.95

Available in COLOR On IDS Prism™ Printer: COLOR PLOT Version $89.95

* HIGH RESOLUTION - 60 x 72 data Points/inch. * LARGE DATA CAPACITY - 1000 Input Data Points per graph. * SELECTABLE GRAPH SIZES - From 1" sq. to 7" x 24" * STANDARD DATA SOURCE - Plots Data from VISICALC or USERS OWN PROGRAMS using the DIF™ Standard Format. * GRAPH FEATURE SELECTION - Fill out Pre-formated Form on VISICALC screen or in users own program. * MINIMAL ENTRY REQUIREMENTS - Enter only name of Datafile and location therein of data to be plotted. * MULTIPLE FUNCTION GRAPHS - Plots over 10 Data Sets per graph. * DATA SYMBOLS - Plots data with user composed symbol shapes. * DATA INTERPOLATION - connects data points with user composed line shapes. * LINE/SYMBOL LIBRARY - Plots each Data Set with different line/symbol shape chosen from 12 line library. * CUSTOM LINES AND SYMBOLS - Has interactive screen-graphics program for composing symbol shapes. * AUTO SCALING - Selects scale values for ease of graph interpretation. User adjustable Mantissa Table. * GRID SELECTION - Prints selectable number of vertical and horizontal grid lines. * CALENDAR SCALE - Optionally prints names of month on horizontal scale. * CURVE SELECTION - Plot each data set with Linear, Stair-Step, or Bargraph curves. * OPTIONAL MIN/MAX VALUES - Extends graph beyond the values of the Data Sets. * DATA SET DESCRIPTIONS - Prints text descriptions of each Data Set in graph legend. * TEXT ENTRYS - Prints graph title, axis labels, and date on graph. * USER FRIENDLY - Checks validity of input data and displays cause of errors. * COMPLETE DOCUMENTATION - Comprehensive 75 page Users Manual with examples covering data preparation, graph feature entry, composing lines and symbols, and technical notes.

USER REQUIREMENTS

COMPUTER
- TRS-80 MODEL I 48K
- TRS-80 MODEL III 48K
- LNW80 48K

DOS
- TRSDOS 1.3, 2.3
- NEWDOS, NEWDOS/80
- DOSPLUS 3.4, LDOS 5.1

DISK DRIVES
- SINGLE DRIVE (NOT TRSDOS)
- DUAL DRIVE (PREFERRED)

GRAPHICS PRINTER:
- MX-80 GRAFTRAX, OR GT +
- MX-100
- LINEPRINTER VIII, DMP 200
- NEC 8023 A-C, C.ITOH 8510
- IDS 460/560, 480, 80/132
- OKIDATA 82/83

++ OTHER VERSIONS IN DEVELOPMENT

TO ORDER: Send check, purchase order, or request for COD shipment. Specify Computer and Printer Type. Include $2.50 for postage and handling. Calif. residents add 6% tax.

MICRO SOFTWARE SYSTEMS • MICROPLOT, INC.

DEALER INQUIRIES WELCOME

1815 SMOKEWOOD AVE. • FULLERTON, CA 92631 • (714) 526-8435

VISA MasterCard VISA/MASTERCARD ACCEPTED

Requested info 7/6/83

in the operator column of the source listing. The operand column contains the location to be loaded and, separated by a comma, the item to be loaded there. Don't let all of this confuse you. Next time, we'll give you a formal introduction to EDTASM.

Put the Number 38 in this Address
English: Put a 26H in the address pointed to by HL.
Object Code: 36 26
Source Code: ASCII LD (HL),26H

The object code 36H means "put the next byte in the address pointed to by HL." When we want HL to show us a RAM address, we write the place to put it in source code as (HL). 26 is, of course, the hex equivalent of 38. That word "ASCII" in the source code is in the label column, which precedes the operator column. It has no effect upon the instruction, but it is a way of referring to this address. We're going to use it later.

Increase SA by One
English: Increase HL (by 1).
Object Code: 23
Source Code: INC HL

Yes, 23H is object code for "add one to HL." By now, you must have been impressed by the incredible simplicity of machine code. Many of the over seven hundred Z-80 instructions are just one byte long. None are more than four bytes. The source code should be self-explanatory.

Has SA Reached 16384?
English: (Step 1) Put the contents of H in the A register.
(Step 2) What happens if we subtract 40H from A?
Object Code: (Step 1) 7C
(Step 2) D6 40
Source Code: (Step 1) LD A,H
(Step 2) SUB 40H

This needs some explaining. Comparison between two integers, i.e., two-byte numbers, is messy in machine language. It's much easier to compare single bytes. How can we compare single bytes and find out if HL has reached 4000H (16384 decimal)? The most significant byte, meaning register H, of the screen memory addresses, will be either 3C, 3D or 3E. When this byte becomes 40H, we have passed the screen memory. So we test H to see if it has reached 40H.

This requires the use of a new Z-80 register, register A. This is the hardest-working register of all. When you are going to do any of the endless things that can be done with a single byte, it has to be placed in register A. Think of it as the byte "operating table." The single byte, 7C, means "copy the contents of register H to register A." Now, if we subtract 40H from A and the result is zero, we'll know that HL is beyond screen memory and it's time to stop. D6 means "subtract the next byte from register A."

If So, Go To Step 60
English: If the zero flag has been set, jump to address 700FH.
Object Code: CA 0F 70
Source Code: JP Z,LOOP

Here, we learn about (but don't actually see) register F, the flag register. Think of it as the messenger, telling concerned relatives what happened to the poor byte on the operating table. Like all bytes in the Z-80, it has eight bits. When the result of an operation on A is zero bit six, the zero flag of the F register is set, turned "on," or becomes 1. The object code (CA) says, "If bit six of the flag register is set, then jump to the address indicated by the next two bytes. These bytes, as always, appear in low/high sequence.

How do we know the correct address for the loop is 700FH? Well, I cheated. I laid all these bytes out starting at 7000H, or 28672 decimal. That made it easy to see that the loop began at 700FH, or 28687 decimal. That's something else you never have to do with source code. You can name the loop instruction LOOP (or any word of six letters or less) and write your source statement as it appears above. EDTASM will figure out the correct address no matter where in memory you have put the program.

If Not, Go Back and Put In Another 38
English: Jump to 7003H
Object Code: C3 03 70
Source Code: JP ASCII

Yes, I cheated here, too, and found that the place where we loaded the 26H into screen memory was 7003H. The source code would have known because we labelled that instruction ASCII. Remember? C3 is object code for JUMP, which is identical to GOTO in BASIC, except that it sends the program to a RAM address and not to a numbered statement.

Keep Looping
English: Jump to this same instruction.
Object Code: C3 0F 70
Source Code: LOOP JP LOOP

There it is. Nothing new here. Just another JUMP instruction that sends the computer around in a tight little circle. The loop can only be broken with the reset button. Remember, the BASIC version required the break key to stop it? Better not have the expansion interface on when you run this program.

This program consists of eighteen bytes — count 'em. How are we going to load it? EDTASM would have written these eighteen bytes from your source code, asked you for a filename and dumped the file to disk or tape. You could then load and run it in the usual way. For now, let's convert these eighteen bytes to decimal and POKE them in by hand, beginning at address 28672 (7000H). Those of you with DEBUG or TBUG can put the hex values in directly. I've taken pity on both of you and put the bytes and addresses in Figure 1. The program won't work anywhere else because it contains addresses which are relative to 28672. That's what we mean by non-relocatable code. Sorry to make you work so hard, but I did say you were going to write this thing. Why not make yourself a DATA, FOR . . . NEXT, READ . . . POKE program in BASIC to do the job? I'll wait for you.

Now is the moment of truth. Do you want to load last month's monitor and check your bytes? When you're ready, command SYSTEM and respond with /28672. If

all went well, you got a screen full of &'s in 0.024 seconds. RESET out of this and have some fun. Simply POKE any ASCII you wish into address 28676 and the program will fill the screen with the equivalent character. Use 191 and get the old chestnut, "whiteout."

This may have seemed complicated. We purposely made it so to give you an intimate understanding of what goes on. If you're still interested in machine language, run out and get yourself one or another version of EDTASM. Join us next time. Things get easier (and more exciting) from here on.

Listing 1 — In the Chips

```
5 REM - MAKE SA POINT TO THE FIRST SCREE
N ADDRESS
10 SA=15360
15 REM - PUT THE NUMBER 38 IN THIS ADDRE
SS
20 POKE SA, 38
25 REM - INCREASE SA BY ONE
30 SA=SA+1
35 REM - HAS SA REACHED 16384?
40 IF SA=16384 THEN 60
45 REM - IF NOT, GO BACK AND POKE ANOTHE
R 38
50 GOTO 20
55 REM - KEEP LOOPING
60 GOTO 60
```

Figure 1

DECIMAL		HEXADECIMAL	
Address	Byte	Address	Byte
28672	33	7000	21
28673	00	7001	00
28674	60	7002	3C
28675	54	7003	36
28676	38	7004	26
28677	35	7005	23
28678	124	7006	7C
28679	214	7007	DE
28680	64	7008	40
28681	202	7009	CA
28682	15	700A	0F
28683	112	700B	70
28684	195	700C	C3
28685	03	700D	03
28686	112	700E	70
28687	195	700F	C3
28688	15	7010	0F
28689	112	7011	70

Captain 80

and the magic software machine

Bob Liddil, Contributing editor

In the last issue, a malfunction in Max's dimensional download system brought a Minotaur into real-time. Being pounded into secret agent burger in the wee hours of the morning in Peterborough, New Hampshire, by a mythical beast almost blew my disguise as a mild-mannered software reviewer. So, Max downloaded us both (Minotaur and me) into the TRS-80 Color Computer which had been recently installed in the lab. Ambushed by the Minotaur, and weaponless, I escaped through a hole in RAM only to find myself a prisoner in the depths of the 6809 microprocessor. My guards are Phantoms, recruited from Phantom Slayer by Med Systems and my captor is none other than Twitch. He is the notorious (but popular) mutated arcade game turned sentient, who has somehow escaped from the ROM cell where he was serving a life sentence for unspeakable crimes against the micro world.

And you thought you'd had a bad day.

"So, human," sneered Twitch, his face twisted in a snarl, "you have fallen into my power once more, and nobody is around to bail you out."

"I wouldn't be so sure of myself if I were you, Twitch," I said with as much confidence as I could muster.

"Master, to you, bit-breath," growled one of the Phantoms in a gutteral tone.

Before I could respond, Twitch motioned him to be silent.

"It does not matter," he said softly. Then, in a booming voice, "Hear my decree. This interloper shall be de-rezzed for crimes against the monarchy. Take him away to the death chamber!"

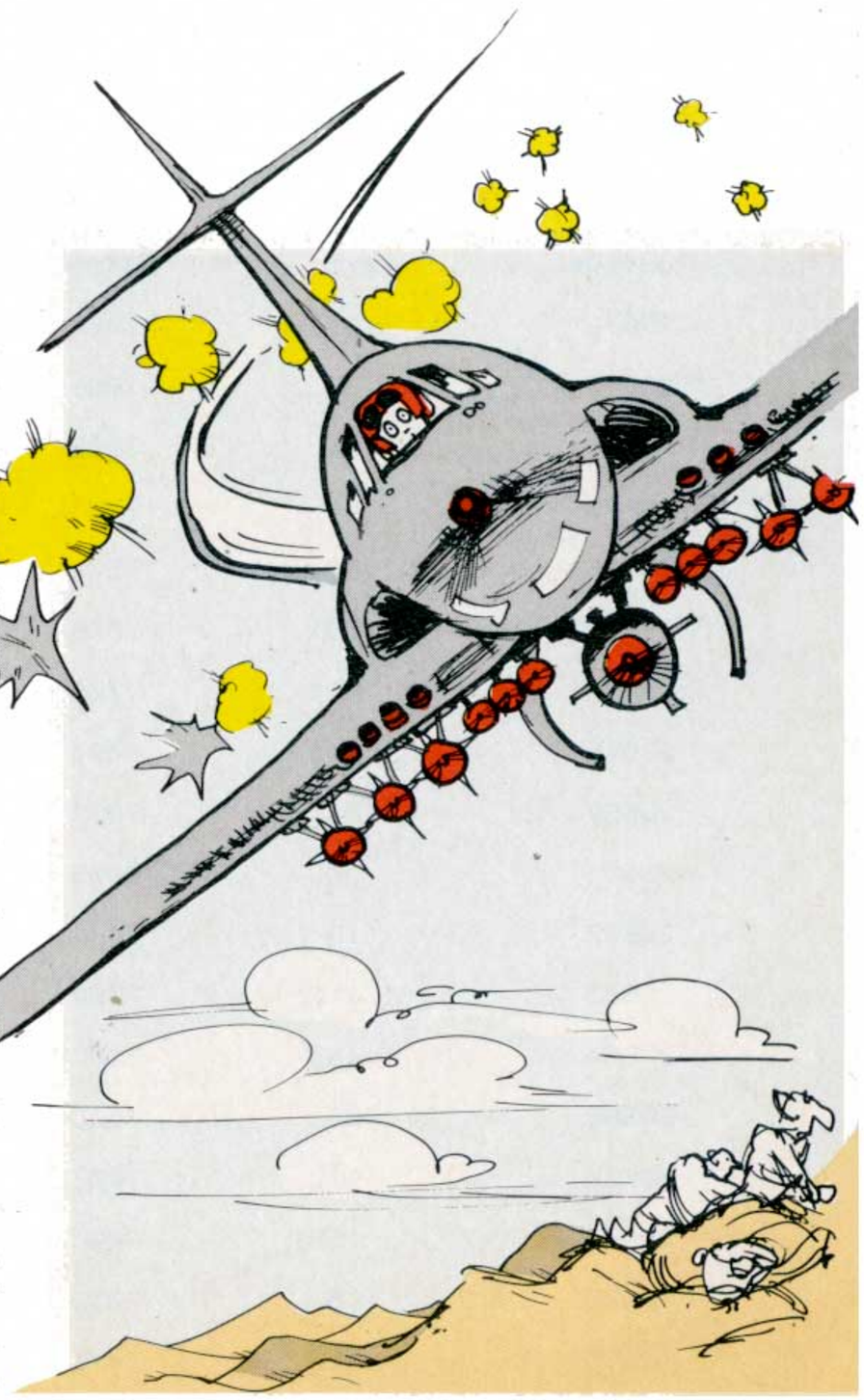
As they led me out of the throne room, I noticed shapes following along on a parallel course. On the plaza outside, the guards flanked me closely, but between them, I caught a better view of those in the periphery. One was decidedly human — dressed in leather armor and carrying a laser pistol at his side. I could read his intentions in his eyes and, as he made his move, I fell to the floor.

The Phantoms' attention was riveted to me for just the nanoseconds required for the youth's device to spray the area with deadly scarlet fire.

Someone grabbed me and dragged me out of the circle of distracted guards. The last thing I saw before darkness overcame me was my benefactor mowing down Phantoms left and right as they closed in on him.

When I regained consciousness, I was sitting in the pilot's seat of an Avenger Mark VI Stinger. It is one of the most versatile of the armed fighter-bombers in all Color Computerdom. My straps had been pulled taut and the same laser-wielding youth was punching equations into my auto-pilot.

"We're sending you back into RAM, Captain," he said through clenched teeth. Say, *how* does he know who I am? Then, quickly, he



ran me through the gunnery procedures.

A hail of laser fire streaked the air. The youth returned the fire and punched the computer control guidance system activator. My Avenger shuddered to life as the youth leapt from the wing, rolling and firing.

"We'll meet again, Captain," he shouted as my craft moved forward, the cockpit closing down around me. The last I saw of him, he was raking his adversaries with red-hot energy bolts while running for a hangar. They were hot on his tail.

I didn't have time to ponder his fate, grateful though I was for his help. With laser eyes aglow, the Avenger was making tracks toward a flight of winged horrors. The weapons systems were locked on manual.

Avenger is a Cornsoft program of such intensity that the player finds himself reaching for the coin slot. This offering is unabashedly Defender, the arcade classic. No attempt beyond the name change has been made to hide that fact. And, true to its heritage, Avenger is a thoroughbred.

The playing field is clean and easy to use. The ship is maneuverable. The enemies are well-defined. It is an arcadeophile's game.

For the player, Avenger gives action in high resolution. There are excellent graphics techniques that stretch the computer to its edges, and an exciting format that entertains.

For the retailer, Avenger offers bright, colorful packaging and (as always with Cornsoft) positive loading for hassle-free customer relations.

I like Avenger. It doesn't task me mentally, or physically, to play it. It's fun and that's enough for me.

It is like a long, terrible roller coaster without rails or anything solid to grab hold of. The monsters which attack me are evil incarnated; minions of Twitch meaning to kill me if they can. I fire. I fire again and yet they continue, wave after wave. I blast my way through them only to be faced with still more.

I find my escape route, a hole in the sky, but my ship is programmed to continue the battle. I rap the flight computer with my fist. It sputters and dies in a puff of smoke. The Avenger streaks for the deck. I grab the stick and wrestle for control.

Then I see her.

She is standing on a knoll, surrounded by monsters. She looks human. I must rescue her. Streaking low along the hilltops, I blast the enemy with a precision unthinkable for a rookie pilot. I scoop her up into the cockpit with me. My lasers are white hot from the strain of combat. I fire again and again, shattering the attackers into molecules.

I bolt skyward, aiming at the hole in the heavens. Cut off, I fire at the attackers until my hand crushes the stick. The button malfunctions and my laser is still.

The hole is above me. I don't yet know who she is, but she trusts me. I reach for the ejection handle, aiming the Avenger at the center of the last wave of the enemy. We pass under the hold and I yank the handle. There is an explosion followed by a thunderous roar. Through the smoke and flame, I see Twitch leering through the waves of heat — tall as a building, yet close enough to touch.

The momentum of the ejection carries us upward. The light is gone but Twitch is shadow and the echo of his scorn follows us upward, fading as the darkness deepens until there is only the deepest black . . . then nothing at all.

I found myself staring upward into the face of Officer B. Goode, of the municipal constabulary. It was drizzling rain and I was freezing.

"You've taken a nasty fall, Bob," Officer Goode was saying. "You really ought to do something about those porch steps."

But I knew better. I could still see her face — her touch lingered. If I was here, where was she? Did she exit into real-time with me? There were questions I wanted to ask Max.

As they lifted me onto the stretcher to put me into the ambulance, a couple of the bystanders caught my eye in the flashes of red light. They were hooded and blended with the crowd . . . one of them was Twitch.

Producing TRS-80 music

Hear the harmony of your computer

Model I

Thomas Trojak, Folsom, PA

The following article presents a BASIC program that allows owners of a TRS-80 computer to play standard musical compositions on their machine, via the tape cassette player/recorder, without the need to add hardware to the system. Hopefully, you will find the program particularly simple and convenient to use. For those users whose interest is limited to the production of music, it is only necessary to read the next section on the use of the program. For those who are interested in the methods used in the production of musical tones, a description of the Z-80 assembly language subroutine used is also given.

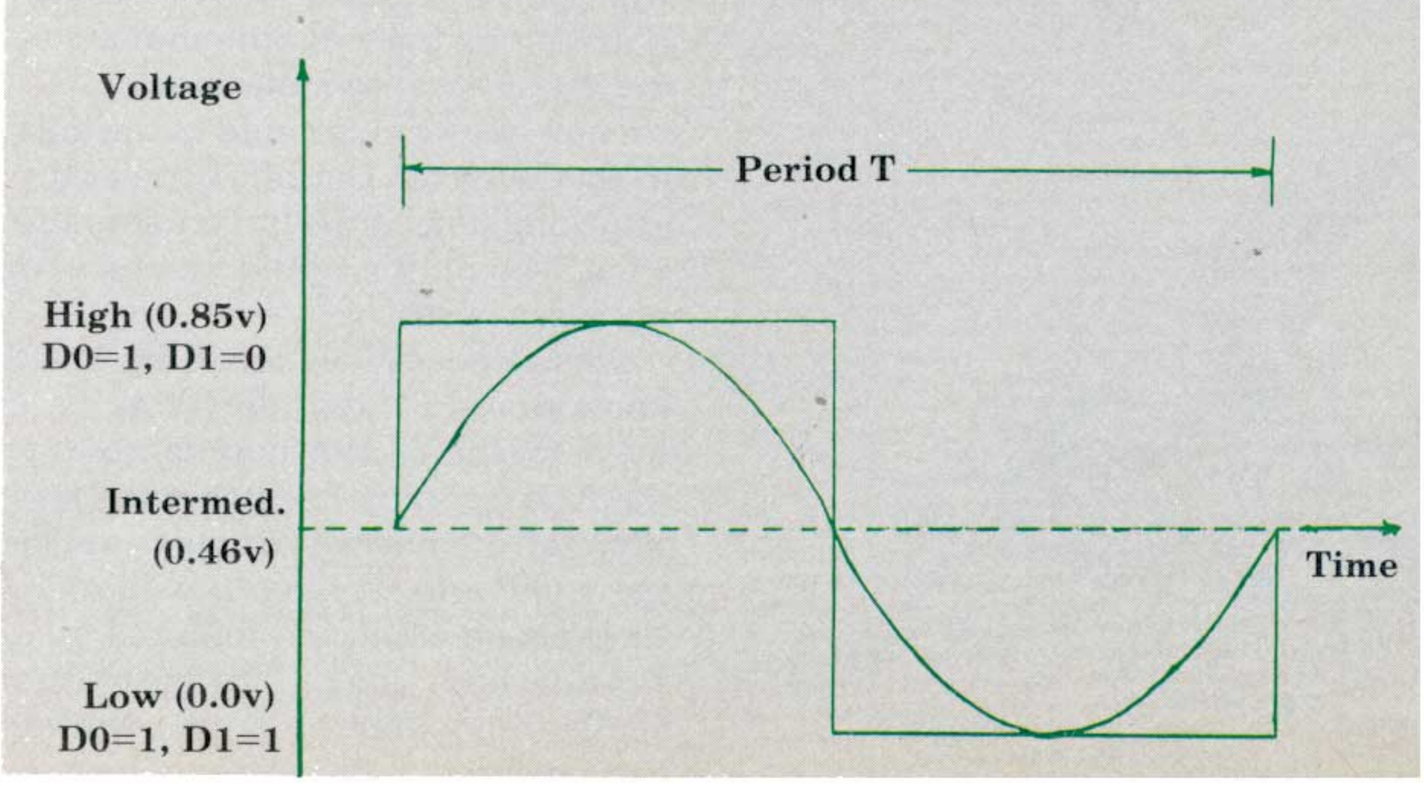
Using the Program

The BASIC program shown in Listing 1 is written for a standard Model I Level II TRS-80 machine. After typing in the program, enter RUN. If a CHECKSUM ERROR appears, you have made a mistake in entering the data of lines 110 to 150. If not, you are ready to enter your composition.

First, the tempo must be set. You are asked to enter the time value, in seconds, that you wish a quarter note to receive. This value must be in the range of 0.2 to 2 seconds. Try entering 0.4 to begin. After this has been done, the program will ask you to enter a note. The following notation is to be used in doing so: The letters C, D, E, F, G, A and B are used to indicate notes within the middle octave (the seven whole tones, beginning with middle C). If a sharped note is desired, the letter is to be immediately followed by the # sign. For example: C#, D#, etc. Flats are not provided for, so they must be converted to sharps.

If notes in the octaves above the middle octave are desired, the notes are to be appended by the number of the octave. For example, entering A1 produces an A in the first octave above the middle octave, while C#3 would produce a C# in the third octave. Notes up to three octaves above the middle octave are available in this program. To produce a rest, enter R. If at any time an unacceptable character string is entered, ILLEGAL will appear on the screen. You will have to reenter that note.

Figure 1: A square-wave approximation to a sine wave of frequency f 1/T, where T is the period. (Actually, it is more accurate to say that the square wave represents a summation of the fundamental frequency f with higher harmonics.)



After you have entered a note, the program asks you to enter the time value of that note. This is done by entering just the denominator of the time value. Enter a 16, 8, 4, 2 or 1 for a sixteenth, eighth, quarter, half or whole note, respectively.

Continue to enter notes and their time values in response to the prompts until the last note of your composition has been entered. Then type DONE in response to the prompt.

At this point, you must prepare the cassette player. Place it in the record mode. Remove the plug from the computer that goes to the EAR jack of the cassette deck, and insert the earphone originally supplied with the player into this jack.

If you now press the E key, the program will terminate. Hit any other key and you will hear your song played over the earphone. It may be repeated as many times as you wish. If you wish to record your piece on a tape, suitable for play on any standard cassette player, simply load a fresh cassette into the recorder, remove the plug from the MIC jack, and execute the play option again.

Finally, as an example, the Scott Joplin tune "The Entertainer" (theme to the motion picture The Sting) is presented in Table 1. Use 0.4 seconds as the duration of a quarter note. Enter each note as listed in the table, followed by its time value (shown in parentheses).

Table 1: The notes to the Scott Joplin song "The Entertainer." Set the duration a quarter note is meant to receive (tempo) to 0.4 seconds. Enter the name of each note as listed above, followed by its time value (shown in parentheses).

D(8), D#(8), E(8), C1(4), E(8), C1(4), E(8), C1(2), C1(4), C1(8), D1(8), D#1(8), E1(8), C1(8), D1(8), E1(4), B(8), D1(4), C1(2), C1(4)

D(8), D#(8), E(8), C1(4), E(8), C1(4), E(8), C1(2), C1(4), C1(8), A(8), G(8), F#(8), A(8), C1(8), E1(4), D1(8), C1(8), A(8), D1(2), D1(4)

D(8), D#(8), E(8), C1(4), E(8), C1(4), E(8), C1(2), C1(4), C1(8), D1(8), D#1(8), D1(8), C1(8), D1(8), E1(4), B(8), D1(4), C1(2), C1(4)

C1(8), D1(8), D1(8), C1(8), D1(8), D1(4), C1(8), D1(8), C1(8), E1(8), C1(8), D1(8), E1(4), C1(8), D1(8), C1(8), E1(8), C1(8), D1(8), E1(4), B(8), D1(4), C1(2), C1(8).

Tone Production

In designing a program to play musical compositions, the first problem we are faced with is how do we produce a tone of a given frequency f? This is done by using the OUT instruction available on the TRS-80's Z-80 microprocessor. Basically, we hold port 0FFH (the cassette port) high for half a cycle and low for the other half. The result is a square wave approximation to a sine wave (see Figure 1). By doing this repeatedly, for the proper length of time, we can produce a tone with the desired frequency, or any duration.

Lines 230 to 320 of the assembly language subroutine in Listing 2 accomplish this. Register H normally contains the value 2. Register C *always* contains 0FFH. When line 230 is encountered, the contents of register C are placed on lines A0 to A7 of the address bus. This signal is then decoded by the cassette port decoding circuitry. The contents of register H are then placed on the data bus. A value of 10_2 on data lines D0 and D1 are passed to a data latch which is enabled by the decoding circuit. The output of the latch, after being passed through a resistor network, holds the signal to the cassette at 0.85 volts (high). The loop CYCLE1 is a delay loop that is designed to allow the cassette signal to remain high for half a period, where the period T is related to the frequency f by T=1/f. The time length of the delay loop is determined by the value initially loaded into register B.

The second half of the cycle is produced in a similar manner. At line 270, the contents of the L register, which normally contains the value 1, is placed on the data bus. A value of 01 appearing on data lines D0 and D1 results in a signal of 0.0 volts (low) being sent to the cassette. The delay loop CYCLE2 allows the signal to remain low for half a period during the second half of the cycle.

Upon exiting the CYCLE2 loop, the contents of register D is decremented. If not zero, the program loops back to INNER. The value initially loaded into register D is chosen such that, for a given frequency, a note with duration of 0.05 seconds is produced by stringing together 0.05/T full cycles. Notice that the value loaded into register D is consequently frequency-dependent.

Now that we have a means of producing a note of a given frequency and duration of 0.05 seconds, we can produce notes that our integer multiples of this time-length by using the loop OUTER. The value of this multiple is initially loaded into the E register.

Timing Considerations

The next problem we are faced with is, for a note of given frequency and duration, how do we determine what values to use in our delay loops? The value that is to be initially loaded into register B must be chosen such that the time it takes the processor to execute the loop INNER *one* time must be equal to the period corresponding to the desired frequency. Most manuals on the use of the Z-80 processor list the execution times (ET) of the various instructions. Using the TRS-80 Editor/Assembler manual, we find that the OUT and LD instructions at lines 230, 240, 270 and 280 have ET's that total to 15.5 microseconds for a processor running at 4MHz. The DEC and JP instructions at lines 310 and 320 increase this by 3.5 microseconds, for a total of 19.0 microseconds. The CYCLE 1 and CYCLE 2 loops each require 3.5Mk microseconds to be executed at 4MHz, where k is the value initially loaded into the B register (i.e., the number of times each of the loops is executed). The total for both loops is then $7.0\mu s \times k$.

Consequently, the time required for one passage through the loop INNER is $(7.0 \times k - 19)\ 10^{-6}$ seconds for a processor running at 4MHz. Now the TRS-80's Z-80 is clocked to run at 1.774MHz., so that the above expression must be multiplied by the ratio 4/1.74 = 2.255. Making this correction, and setting the time required for one passage through INNER equal to the period T, we

IJG DEALERS HAVE TRS-

Computer Books and Software for the TRS-80s, are at your IJG Dealer Today.

Books

TRS-80 Disk & Other Mysteries.
The "How To" book of data recovery by H. C. Pennington. 128 pages, $22.50

Microsoft BASIC Decoded & Other Mysteries.
The complete guide to Level II operating systems & BASIC by James Farvour. 312 pages, $29.95

BASIC Faster & Better & Other Mysteries.
Microsoft BASIC programming tricks & techniques by Lewis Rosenfelder. Software available on disk. Radio Shack Cat. No. 62-1002. 290 pages, $29.95

The Custom TRS-80 & Other Mysteries.
A guide to customizing TRS-80 hardware and software by Dennis Bathory Kitsz. Schematics and listings. 336 pages, $29.95

NEW

TRSDOS 2.3 Decoded & Other Mysteries.
The TRSDOS operating system explained by James Farvour. Disassembly of code with commentary. 300 pages, $29.95

BASIC Disk I/O Faster & Better & Other Mysteries.
Programming techniques and helpful subroutines, by Lewis Rosenfelder, for BASIC programs which store or retrieve data from disk. (Available in June) $29.95

NEW

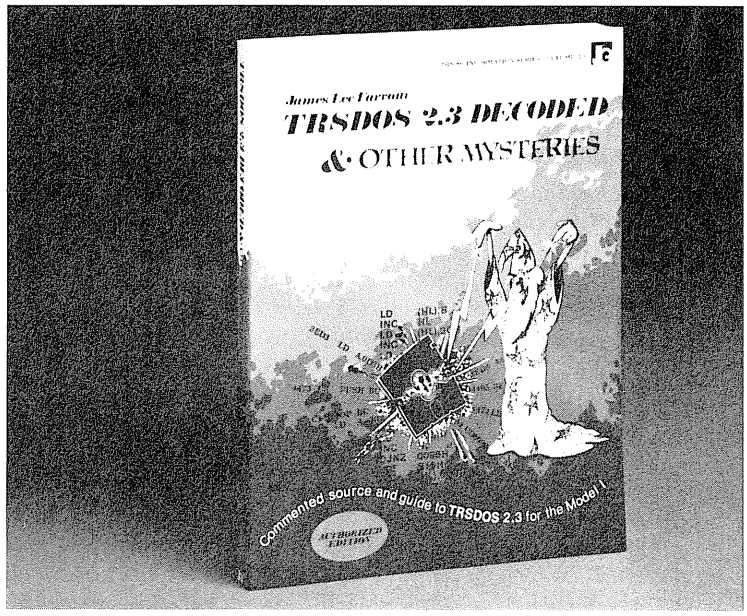
How To Do It On The TRS-80.
The applications guide to the TRS-80 Models I, II, III & Color Computer by William Barden, Jr. 300 pages, $29.95 (Available in Spring '83)

Machine Language Disk I/O & Other Mysteries.
The guide to machine language disk software for TRS-80 Models I & III by Michael Wagner. 288 pages, $29.95

Electric Pencil Operators Manual.
By Michael Shrayer and H. C. Pennington. 123 pages, 24.95

The TRS-80 Beginners Guide To Games & Graphics.
Simple programs teach basic concepts of graphics and game design, by Tom Dempsey. (Available in July) $24.95

The Captain 80 Book of BASIC Adventures.
Eighteen program listings plus Adventure program generator, by Bob Liddil. 252 pages, $19.95



NEW BOOK!

Software

Electric Pencil 2.0z Word Processing System.
The easy to learn, easy to use word processing system by Michael Shrayer. Includes operators manual. Disk $89.95, Cassette or Stringy Floppy $79.95

NEW

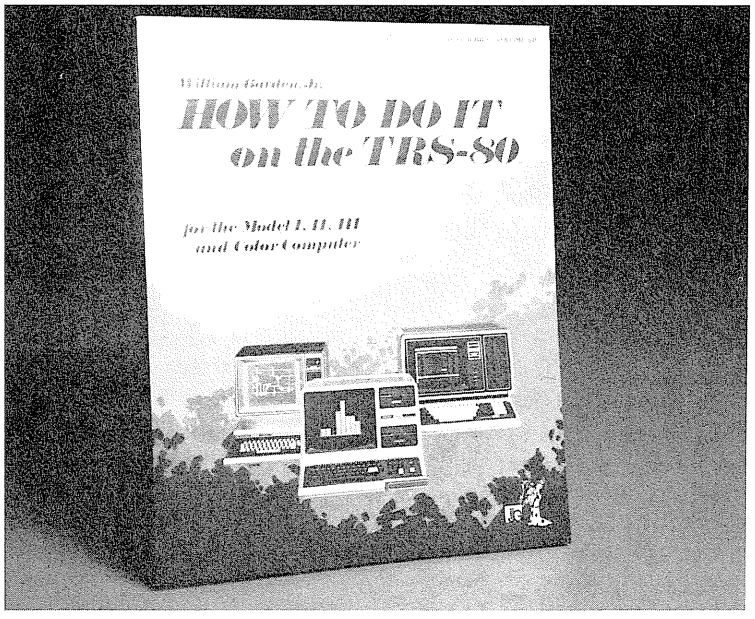
BLUE Pencil — 50,000 Word Expandable Dictionary. Companion to the Electric Pencil 2.0z word processing system. Disk $89.95

NEW

RED Pencil — Automatic Spelling Correction. For use with the Electric Pencil 2.0z word processing system. Must be accompanied by Blue Pencil to operate. Disk $89.95

BFBDEM — BASIC Faster & Better Library disk by Lewis Rosenfelder. 32 demonstration programs, BASIC overlays, video handlers, sorts and more for the Model I & III. Radio Shack Cat. No. 260-2021. Disk only $19.95

80® BOOKS & SOFTWARE.



NEW BOOK!

BFBLIB — BASIC Faster & Better Demonstration disk by Lewis Rosenfelder. 121 functions, subroutines and user routines for the Model I & III. Disk Only 19.95

Utilities

TRANSLATE Convert any character to any character or string. Create your own shorthand. Print special characters. Disk $49.95

DISKMAP Produces two different reports; a listing of disk space allocation by granule, and a listing of all granules allocated to each data file. Disk $29.95

Games

CYBERCHESS
Chess Improvement System. Not a game, but a powerful and effective method for improving one's skill in chess. Choose from 55 amateur or 55 professional disk packets with 4 different games on each. System Disk $29.95 (Includes four games). Each amateur or professional disk $19.95 each.

FLAG RACE
Race your car through a maze and try to reach all the flags before being caught and killed by drone cars. Can you do it? For Models I & III. Disk $24.95

SPACE ROBBERS
Inter-galactic thieves are after your supplies and you must stop them before they take it all. For Models I & III. Disk $24.95

INTERCEPTOR
The aliens are attacking you in wave after wave, can you survive and get back to the mother ship to refuel? For Models I & III. Disk $24.95

ALIENS
Invaders attempt to land and you must stop them. But watch out, you're dead if they land on you. For Models I & III. Disk $24.95

NEW

FUNSOFT Games distributes through IJG for Models I & III.

BABLE TERROR Bables are roaming the maze like complex everywhere but you can only see a few yards ahead, can the Bables be cleared out before they clear out you? Disk $24.95, Cassette $19.95

MAD MINES Mad Mines are being placed into the space around your planet. As their mad pace speeds up, the situation becomes more difficult. Can they all be destroyed? Disk $24.95, Cassette $19.95

APPLE PANIC Crazed apples chase you over many ladder connected brick levels. Your only hope is to dig holes in the floor and beat them down when they get stuck. Disk $24.95, Cassette $19.95

THE BLACK HOLE Your mission is to seek-out and destroy the Dorfian leader. But, can you survive the perils of the Black Hole? Disk $24.95, Cassette $19.95

TIME RUNNER Newly discovered land is yours for the taking. All you have to do is take it . . . before the defender droids catch you that is. Disk $24.95, Cassette $19.95

IJG products are available at computer stores, B. Dalton Booksellers, Radio Shack Computer stores, and independent dealers around the world.

If IJG products are not available from your local dealer, order direct from IJG. Include $4.00 for shipping and handling per item. Foreign residents add $11.00 plus purchase price per item. U.S. funds only please!

IJG, Inc.
1953 West 11th Street
Upland, California 91786
Phone: 714/946-5805

If it's from IJG IT'S JUST GREAT!

TRS-80 TM Tandy Corp.
Microsoft TM Microsoft Corp.

have:

$$T=(15.78k-42.84)\times10^{-6}$$

Solving this equation for K, and using the relation T=1/f, where f is the frequency in Hz., we obtain:

$$K=\frac{(10^6/f-42.84)}{15.78}$$

For a given frequency f, we use this equation to obtain the value of k to be inserted in the B register at the beginnings of the CYCLE1 and CYCLE2 loops. For example, for the note A, F=440Hz., so that the value 141 must be loaded into B. Note that the value of k must be rounded to the nearest integer for use by the machine language subroutine. The actual calculation of k for a given frequency is done at line 430 of the BASIC program of Listing 1.

The calculations of the values to be used as counters in loops INNER and OUTER are relatively straightforward. For the inner loop to produce a tone of 0.05 seconds, we must load register D with the value 0.05/T=0.05×f. If a total duration of length d (in seconds) is desired, register E (the counter for loop OUTER) must be loaded with the value d/0.05. These two calculations are done in the BASIC program at lines 480 and 470, respectively. The values to be loaded into registers D, E, and B are stored, in that order, in consecutive locations of memory, starting at location 27850 (directly following the machine language subroutine). The POKE statements at lines 480, 470 and 440 accomplish this.

Additional Comments

This completes the description of the basic workings of the assembly language program. A few additional comments may be useful, however.

First, lines 130 to 150 of Listing 2 make it possible to return to the BASIC program by inserting a value of zero into the first location of the three-byte sequence needed to produce a note.

Lines 170 to 190 allow for the possibility of producing *rests* in the musical composition. Normally, the H and L registers are loaded with 2 and 1, respectively. If, however, a zero is encountered in the third location of the three-byte sequence, H and L are both set to zero. When the OUT statements of lines 230 and 270 are encountered, 00_2 appears on data lines D0 and D1, the result being that the cassette signal is held at a steady value of 0.46 volts (intermediate). As a consequence, no tone is heard. A "rest" is produced.

Lines 350 to 400 produce a slight pause between notes, to prevent a "running together" of notes. Lines 410 to 440 cause the program to jump to the next note.

The BASIC Program; Description

Lines 1 to 150 POKE the assembled machine language subroutine contained in the DATA statements into memory locations 28672 to 28749.

Lines 300 to 440 ask the user to input an ASCII string, interprets the string, and prepare it for use by the machine language routine. The string is first checked to see if a DONE or R (rest) have been encountered, acting appropriately if so. If not, the ASCII representation of notes is converted to the proper frequency. Lines 340 and 360 convert the first character, which must be an A, B, C, D, E, F, or G, to its corresponding frequency in the middle octave. If a # sign is then encountered in the string, line 380 multiplies this frequency by the value $2^{1/12}=1.0595$. (This is a general property of equally-tempered scales.)

Finally, if a number n is encountered, indicating the number of octaves above the middle octave, the frequency is multiplied by 2^n to produce the desired frequency (another general property of scales).

Lines 430 and 440 convert the frequency into data suitable for use by the machine language routine, poking the data into memory, beginning at location 28750.

Lines 450 to 480 ask for the duration of the note and prepare the data, poking it into memory.

Lines 490 and 500 increase the counter K and cause the program to ask for the next note.

Lines 600 to 999 allow the user to play his composition or end the program.

As a final note, it may interest the reader to realize that once a composition has been entered, it exists in a data table beginning at location 28750. The machine language subroutine and data table can then be saved as object (SYSTEM) code. Or, if the piece is not too long, one might consider PEEKing at the values, and then appending them to the DATA statements of the BASIC program. In this way, this program could be used to create subroutines that play a specific piece, suitable for use within other programs. Perhaps you have a game program in mind that could use a music routine. At any rate, I hope you enjoy using this program as much as I have.

Listing 1: The BASIC program that allows the user to enter notes to a musical composition, then play them on a TRS-80 microcomputer.

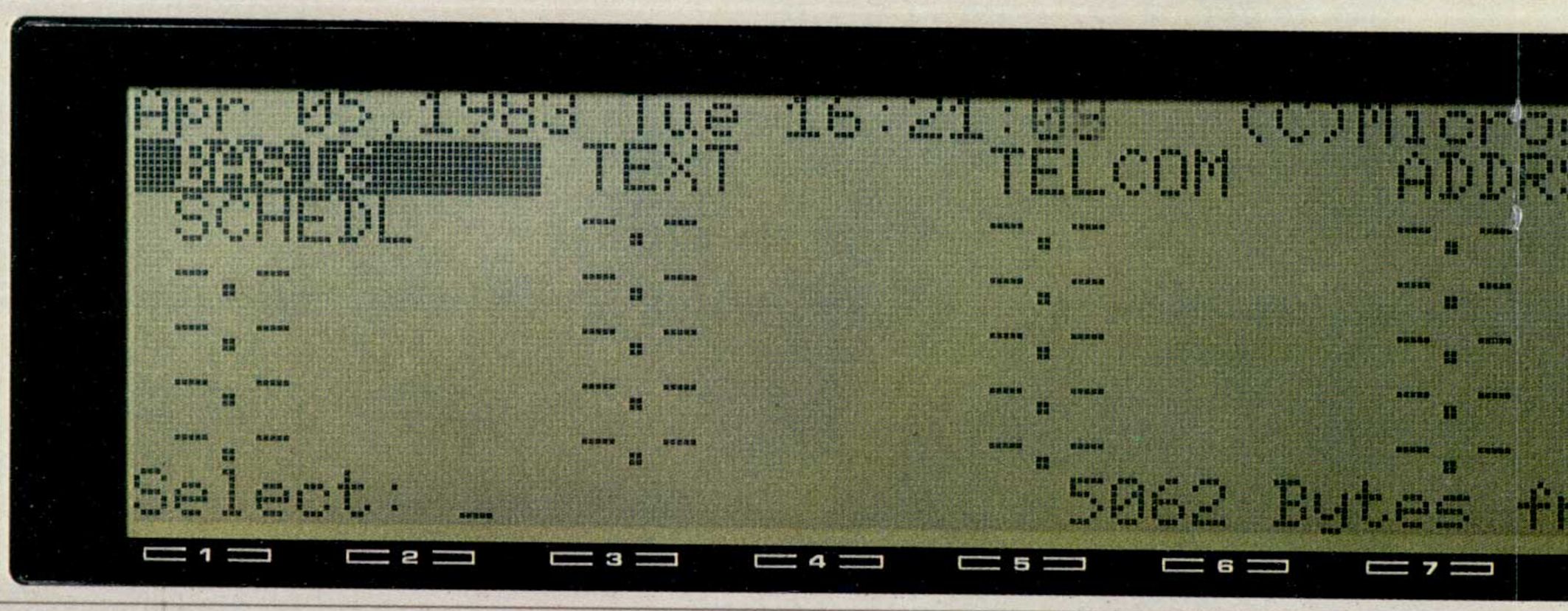
```
1 CLEAR 1ØØØ: POKE 16561,2
55: POKE 16562,111: CLEAR
1ØØ
2 POKE 16526,Ø: POKE 16527
,112: CLS
1ØØ FOR X=28672 TO 28749:
READ D: CS=CS+D: POKE X,D:
 NEXT
11Ø DATA 221,33,78,112,14,
255,221,126,Ø,183,2ØØ,33,1
,2,221,126
12Ø DATA 2,183,194,24,112,
33,Ø,Ø,221,94,1,221,86,Ø,2
37,97,221
13Ø DATA 7Ø,2,5,194,35,112
,237,1Ø5,221,7Ø,2,5,194,44
,112,21,194
14Ø DATA 3Ø,112,29,194,27,
112,62,Ø,211,255,17,255,33
,64,Ø,25
15Ø DATA 218,66,112,221,35
,221,35,221,35,195,6,112
16Ø IF CS<>8Ø78 PRINT "CHE
CKSUM ERROR": END
2ØØ PRINT "ENTER LENGTH OF
 QUATER NOTE IN SECONDS"
21Ø INPUT "RANGE MUST BE B
ETWEEN .2 AND 2";L:CLS
22Ø IF L<.2 OR L>2 PRINT "
OUT OF RANGE": GOTO 2ØØ
23Ø L=INT(L/.2): L=L*.2
24Ø FOR I=1 TO 7: READ F(I
): NEXT
25Ø DATA 44Ø, 493.9, 261.6
, 293.7, 329, 349.2, 392
```

```
300 INPUT "ENTER NOTE";N$
310 IF N$="DONE" POKE 2875
0+K,0: GOTO 600
320 IF N$="R" FR=1000: POK
E 28752+K,0: GOTO 450
330 IF N$="" PRINT "ILLEGA
L": GOTO 300
340 I=ASC(N$)-64
350 IF I<1 OR I>7 PRINT "I
LLEGAL": GOTO 300
360 FR=F(I): IF LEN(N$)=1
GOTO 430
370 N$=MID$(N$,2)
380 IF ASC(N$)=35 FR=FR*1.
0595 ELSE GOTO 400
390 IF LEN(N$)=1 GOTO 430
ELSE N$=MID$(N$,2)
400 V=VAL(N$): IF LEN(N$)<
>1 PRINT "ILLEGAL": GOTO 3
00
410 IF V>0 AND V<=3 FR=FR*
2[V: GOTO 430
420 PRINT "ILLEGAL": GOTO
300
430 C=(1E6/FR-42.84)/15.78
440 D3=INT(C+.5): POKE 287
52+K,D3
450 INPUT "ENTER TIME VALU
E";D
460 IF D<1 OR D>16 PRINT "
ILLEGAL": GOTO 450
470 D2=80*L/D: POKE 28751+
K,D2
480 D1=INT(.05*FR+.5): POK
E 28750+K,D1
490 K=K+3: PRINT
500 GOTO 300
600 CLS: PRINT "INSERT EAR
PHONE INTO RECORDER"
610 PRINT "SET IN RECORD M
ODE": PRINT
620 PRINT "PRESS 'E' TO EN
D, ANY OTHER KEY TO PLAY"
630 A$=INKEY$: IF A$="" GO
TO 630
640 IF A$<>"E" A=USR(0): G
OTO 630
999 END
```

Listing 2: The Z-80 assembly language program that produces notes of a given frequency and duration through the TRS-80's cassette port.

```
7000            00100          ORG     7000H
7000 DD214E70   00110          LD      IX,704EH        ;POINT TO START OF DATA
7004 0EFF       00120          LD      C,0FFH          ;ADDRESS OF CASSETTE PORT
7006 DD7E00     00130 NEW      LD      A,(IX)          ;TEST FOR END
7009 B7         00140          OR      A
700A C8         00150          RET     Z               ;IF END, RETURN TO BASIC
700B 210102     00160          LD      HL,0201H        ;SET H&L TO 1&2
700E DD7E02     00170          LD      A,(IX+2)        ;TEST FOR REST
7011 B7         00180          OR      A
7012 C21870     00190          JP      NZ,NOREST       ;IF NO REST, JUMP
7015 210000     00200          LD      HL,0000H        ;IF REST, SET H&L TO ZERO
7018 DD5E01     00210 NOREST   LD      E,(IX+1)        ;OUTER LOOP COUNTER
701B DD5600     00220 OUTER    LD      D,(IX)          ;INNER LOOP COUNTER
701E ED61       00230 INNER    OUT     (C),H           ;PORT GOES HIGH
7020 DD4602     00240          LD      B,(IX+2)        ;FREQUENCY COUNTER
7023 05         00250 CYCLE1   DEC     B               ;DELAY FOR 1/2 PERIOD
7024 C22370     00260          JP      NZ,CYCLE1
7027 ED69       00270          OUT     (C),L           ;PORT GOES LOW
7029 DD4602     00280          LD      B,(IX+2)        ;FREQUENCY COUNTER
702C 05         00290 CYCLE2   DEC     B               ;DELAY FOR 1/2 PERIOD
702D C22C70     00300          JP      NZ,CYCLE2
7030 15         00310          DEC     D               ;DECREMENT INNER COUNTER
7031 C21E70     00320          JP      NZ,INNER        ;END OF INNER LOOP
7034 1D         00330          DEC     E               ;DECREMENT OUTER COUNTER
7035 C21B70     00340          JP      NZ,OUTER        ;END OF OUTER LOOP
7038 3E00       00350          LD      A,0             ;PAUSE BETWEEN NOTES
703A D3FF       00360          OUT     (0FFH),A        ;PORT GOES INTERMEDIATE
703C 11FFFF     00370          LD      DE,-1
703F 214000     00380          LD      HL,40H          ;COUNTER FOR PAUSE
7042 19         00390 PAUSE    ADD     HL,DE
7043 DA4270     00400          JP      C,PAUSE         ;END OF PAUSE LOOP
7046 DD23       00410          INC     IX              ;POINT TO NEXT NOTE
7048 DD23       00420          INC     IX
704A DD23       00430          INC     IX
704C C30670     00440          JP      NEW             ;JUMP TO BEGINNING
0000            00450          END
00000 TOTAL ERRORS
```

9 on the Richter



er Scale

SHOWN ACTUAL SIZE

Radio Shack's New TRS-80® Model 100 Portable Computer Is Sending Shockwaves Throughout the Industry

8K TRS-80 Model 100

799.00 Cat. No. 26-3801

Only $54 a Month on CitiLine Credit

24K TRS-80 Model 100

999.00 Cat. No. 26-3802

Only $68 a Month on CitiLine Credit

- **Powerful Built-In Software and Extended BASIC Language**
- **Self-Contained Direct-Connect Telephone Modem**
- **8-Line by 40-Character Display**
- **Full-Size Typewriter Keyboard**
- **Retains Data When Power "Off"**
- **Memory Expands to 32K**

Brace yourself—Radio Shack just redefined the concept of personal computing. Our new TRS-80 Model 100 is so small, it'll fit in your in-basket. Yet, it's as powerful as many desktop micros. The built-in BASIC language is the most advanced ever from Microsoft®, and features full string handling, complete file operations, multi-dimension arrays, 14-digit double-precision accuracy, 240 × 64 dot-addressable graphics, a five-octave sound generator and much more. The Model 100 includes a parallel printer port, an RS-232C serial interface and a cassette port. Model 100's built-in direct-connect modem even allows you to access any computer by phone. If you're not a programmer, "instant-on" software in ROM lets you use your Model 100 as a personal word processor, telephone auto-dialer, address book and appointment calendar. Come see the most revolutionary computer since the TRS-80 Model I at over 6500 Radio Shack stores and participating dealers, including over 400 Radio Shack Computer Centers nationwide.

Radio Shack®

The biggest name in little computers®

A DIVISION OF TANDY CORPORATION

Double-width video

Poking around in your computer

Models I/III

Dan Sitch, Pittsburg, NH

Have you ever wanted to escape the double-width mode without clearing the screen? Pressing the clear key or using the CLS statement in your program will, unfortunately, clear the screen and send the cursor home.

By using the POKE and OUT combination with address 4210H (16912), and I/O PORT ECH (236) we can easily escape the double-width mode and not clear the screen. For those of you unfamiliar with how the POKE and OUT statements function, let's take a minute and see what happens when we POKE an address.

Each address can hold one byte. Each byte is comprised of eight bits which can be turned ON or OFF. Each bit has a value of twice the value of the one before it. The 0 bit has a value of one, the 1 bit has a value of two, the 2 bit has a value of 4, and so on, until we get to the number 7 bit, which has a value of 128. The total of all eight bits (0-7) is equal to 255. By POKEing any number 0 to 255, we can turn on (or off) any bit we desire. Below is a chart that may help you better understand how POKE works.

Figure 1

7	6	5	4	3	2	1	0	8 bits (0-7)=1 byte
128	64	32	16	8	4	2	1	Value assigned to each bit. Total= 255
off	off	on	off	on	off	off	off	Bit status value=

3.

on and
pective-
mber 3

and five bits on (in address 16912), all we would have to do is type POKE 16912,40. The computer would see 00101000 and go do its thing.

You can POKE into any part of your RAM, and if you have 64K of memory, you have over 50,000 addresses you can poke to. Before someone jumps all over me, let me *caution* you that if you POKE into certain parts of your RAM, some weird and funny things will happen. Be judicious in your POKEing and your machine will do wonderful things for you.

The OUT statement works pretty much the same as POKE, only it is used to turn on (and off) the port bits of your machine, and you have 256 ports (0-255). Again, *caution* is warranted. Some ports are used for your machine to operate properly. Be very careful when you use the POKE and OUT statements.

Let's get to what this article is about. Power up your machine and do a PEEK at 16912. PEEK is a function that allows us to examine any address and see what value that address is holding. You may PEEK anywhere in memory, be it ROM or RAM, without worrying about your machine going into a black hole. Please do not clear the screen until we are finished with the PEEKs and POKEs.

Okay, let's try again. Type PRINT PEEK (16912) and press enter. A value of 40 should pop up on the screen. Now, type PRINT CHR$(23) and press enter. I know, *big deal*, we're in double width. Again, PEEK at address 16912. The value is now 44. One more POKE and *please* pay attention. Type POKE 16912,40 and press enter. You just escaped the double width and the screen did not clear!

Well! You would reason that all you would have to do is POKE 16912,44 in your program to get into double width and POKE 16912,40 to escape without clearing the screen. Not so! For some reason unknown to me, it will work in the immediate mode, but not in a program.

Don't panic. I'll show you how to use the OUT state-

ment in conjunction with the POKE statement to escape double-width mode in a program and not clear the screen.

First, let's prove that POKE, used alone in a program, will not work. You type and I'll wait:

```
10 CLS                              'clear screen
20 POKE 16912,44                    'go double width
30 PRINT @ 82,"DOUBLE WIDTH"
40 GOSUB 200                        'wait a bit
50 POKE 16912,40                    'back to normal
                                    width
60 PRINT @ 534,"NORMAL WIDTH"
70 GOSUB 200                        'wait a bit
110 GOTO 10                         'do it again
200 FOR W=1 TO 500: NEXT:           'wait & back to
RETURN                              where you come
                                    from
```

Run the program. No double width. Are you sure? Take a close look at the top line on the screen. Notice the spacing. Did you type it in that way? Hit the BREAK key when only the top line (DOUBLE WIDTH) is on the screen. Magic! We have double width. The poor soul was in double width all of the time, but too confused to realize it. Let's help the machine get its head together with the OUT statement. Change lines 20 and 50 to read:

```
20 POKE 16912,44 : OUT 236,4        'real double width
50 POKE 16912,40 : OUT 236,0        'back to normal
```

Add lines 80, 90, and 100:

```
80 CLS
90 PRINT @ 598,"CLEAR SCREEN"
100 GOSUB 200
RUN
```

For suffering with my article, I'll give you a quickie to POKE in address 16912. Type POKE 16912,41 and watch the upper right corner of the screen. Hmmm, could be of some use! Don't ask me how to get rid of it. Read the article one more time if you have to ask that question. You didn't pay attention the first time.

I have not forgotten the Model I. This may be old hat to Model I owners, but for what it's worth, change line 20 and 50 to read:

```
20 POKE 16445,8 : OUT 255,8
50 POKE 16445,0 : OUT 255,0
```

Being a fair sort of fellow, and not wanting the OUT's feelings to be hurt, I'll give you an OUT to try all alone.

Make sure your cassette is hooked up to your machine. Lift the tape compartment lid and depress the play button. Type (you do not need a line number): OUT 236,2 : FOR W = 1 TO 2500 : NEXT. Press enter and watch the LED indicator and the righthand hub.

Now you are able to escape double width without clearing the screen, put something in the upper right corner of the screen, and watch the hub on your cassette whirl.

If you have digested all of that, you have digested quite a bit.

Files and foibles

Going sequential

For all models

Terry R. Dettmann, Associate editor

Quite a few people have written to say they've enjoyed our little forays into file handling. A few have even brought up an area we've largely ignored, sequential files.

I can already see a few of you looking at the page and saying, "Has he gone nuts or something? Sequential files are a dead subject. *Everyone* knows about those!" I hope to show that sequential files have their place as well as random files do. In fact, in some cases they are far more appropriate to the job at hand.

So far in this series we've used sequential files without getting heavily involved with them. Now let's talk about them in more detail.

What is a Sequential File?

To understand a sequential file, you simply have to understand the word sequential. Webster's *New World Dictionary* defines the word as follows:

1: of, relating to, or arranged in a sequence . . .

2: following in sequence.

Information stored in a sequential file is stored so that one thing comes after another.

Unlike a random access file, a sequential file cannot get at something in the middle without first reading everything in between. Many of us started out using magnetic tape for storage (cassette on micros, larger tapes on big systems). Tape gives us a good mental picture of the problems of dealing with a sequential file.

If we have 1000 names stored in a mailing list on tape, they are stored one after another on the tape as shown in Figure 1. Just like a Beethoven symphony or the latest from a popular rock group on tape, if you want to find something in the middle of the tape, you have to search through everything before it to process it. With music we can search faster, but we still have to go through the selections sequentially.

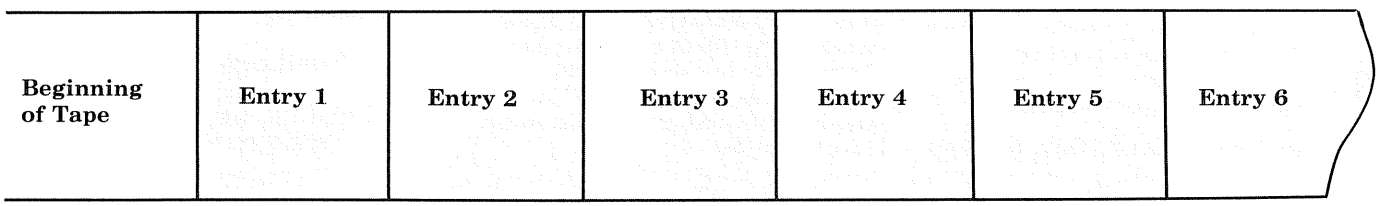
Some very sophisticated professional tape systems used on mini-computers, and larger systems, are able to make rapid jumps into a tape by fast forwarding to a known spot. But even then we are still passing over everything located before what we want instead of just jumping to the right spot.

Now, let's say that I want to correct the 999th entry in my mailing list. If I go to find it, I have to first pass through the first 998 entries, even if I know that the 999th one is the one I want! Understanding this fact is fundamental to using sequential files effectively.

Once I'm at the 999th entry, I have still another problem. What if what I want to put back something that isn't the same length as what I took out? If it's too short, the system will see two entries in place of the one. If it's too long I'll lose part of the 1000th entry.

Dealing with these problems is the essence of the sequential file handling problem. When we use sequential files on a TRS-80, the last problem is circumvented by having the computer prevent you from writing to a file while you are using

Figure 1



it for reading and vice versa.

That's not true of all systems. If you're dealing with files on the Model 16 under XENIX, ALL files are inherently sequential because of the way XENIX is designed. Even random access file structures are built from sequential files.

But let's limit ourselves to dealing with Microsoft BASIC. If we can't read and write to the same file at the same time, we wind up having to deal with two files at the same time; an input file and an output file. Or, we read everything into memory and then process it later out to disk.

The first method limits you in the methods you can use to process a file, the second runs the danger of losing the information altogether if a system failure occurs during processing or while the file is being rewritten. We'll deal with applications of both methods when they are appropriate.

First let's work with the in-memory processing idea. The basic structure of a program that does in-memory processing is shown in Figure 2. It consists of reading in the file information, processing it, and writing it out to the disk again. It sounds simple but there are many variations on this structure, some of which you will find very useful.

Reading a Sequential File

In order to read a file, we have to first establish a link between our program and the file. This is the purpose of the OPEN statement.

Logically, our program sees the file in terms of entries that it is reading from or writing to the disk. On disk, the actual storage forms a magnetic image of what we want to record. In order to establish a connection between the two, the OPEN statement creates an area in memory called a buffer. Each buffer is assigned a number which can be used in our program to refer to it.

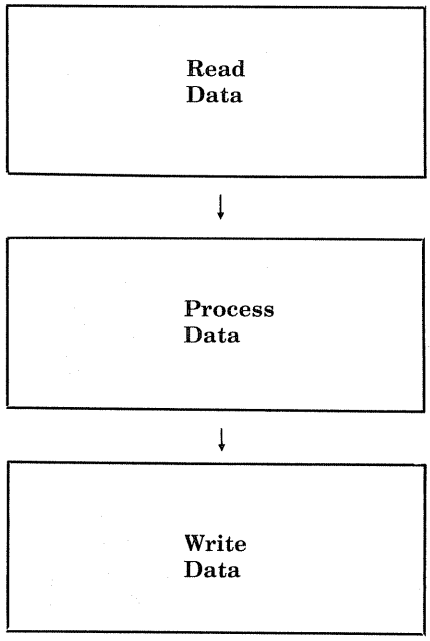
On disk, each file is assigned a name, its filename and possibly an extension used to distinguish between related files or types of files. OPEN connects the two along with something to tell the system whether to expect reading or writing from the file.

The form of the OPEN statement is: OPEN access type, buffer number, filename

The access type for sequential files is allowed to be "O" when we are writing to the file (Output) or "I" when we are reading from the file (Input). Some systems have other access types such as "A" (Append) where the file is opened for output to start at the end of file.

Buffer number refers to which buffer you want to assign for use by the file. On a Model III, when you go into BASIC, the question is asked "How many files?". The number you answer there (default is 3 if you hit enter) is the number of buffers available for files. They are ALWAYS numbered from 1. On the Model II, to get three files, you must include a "-F:3" after BASIC when

Figure 2



starting BASIC because there is no default for the Model II.

Filename is what you have called your data. This must follow all specifications for TRSDOS filenames listed in your TRSDOS manual.

When a file is opened for input ("I") or output ("O"), the file is always positioned so that it is at the beginning (unless you have a system with one of the other access options). When reading from the file, this means we'll always be working from beginning to end. When writing the file, we'll always be rewriting the file. Whatever was there before is lost.

The opposite side of the OPEN statement is the CLOSE statement. In order to disconnect a file from a buffer we say: CLOSE buffer number

This allows us to reuse the buffer for something else or simply make sure that the file is properly closed.

Simple reading

To read information from a sequential file we use the INPUT statement. By itself, INPUT is directed to get information from the keyboard. However, to refer to a file, we simply refer to the buffer assigned to a file by an OPEN statement and we can read from the file.

To refer to a buffer, we add a pound sign (#) and the buffer number followed by a comma to the INPUT statement. This redirects INPUT to use the buffer instead of the keyboard.

For example, if we were reading from buffer number 1 (previously opened with an OPEN statement), then we would write: INPUT#1, . . .

What can we read? We can read anything we might read from the keyboard. If we wanted to read in three numbers, we might use: INPUT#1, A,B,C. If we wanted to read in a string, we might use: INPUT#1, D$. Be careful. If I have three numbers stored on disk *after* a string, reading with: INPUT#1, D$,A,B,C will mess things up because the numbers will be included in the string. I'll explain that later, for the moment just be aware of it.

Writing to Disk

We write to disk the same way we write to the terminal, with the PRINT statement. Like INPUT, it is modified to refer to the buffer we are dealing with. We would say: PRINT#1, A,B,C to write the three numbers A,B, & C to the disk. To write a string: PRINT#1, D$ or the combination: PRINT#1, D$,A,B,C. This last statement will write to the disk without trouble but will lead to the INPUT problem mentioned above.

Next time, we'll start taking these statements and putting them together to get programs that do sequential disk operations.

Micro harmony

Construct four-part harmony on Model I

Model I

Ray Bennett, Seattle, WA

Have you ever dreamed that your TRS-80 Model I could put you to sleep with a lullaby, or serenade your sweetheart, or convince your music teacher that you are practicing your organ lessons (and improving) while you go fishing? With the addition of Micro-Harmony, you can do all of these and much more (well, maybe not fool your music teacher).

Micro-Harmony is a blend of hardware and software — neither being capable of anything without the other. The hardware portion of it will do more than play music, as we shall see, but its main emphasis is to play four-part harmony. The hardware (nuts, bolts and wires) is actually capable of six music parts, but the software under BASIC isn't fast enough to keep up. It's an excellent application for a compiler, but that's another subject.

The music produced comes out at a level compatible with "AUX" inputs on amplifiers and tape recorders. The output is lightly filtered to remove some of the harsh overtones that go with a square wave. As a result, the sound is quite pleasant, being reminiscent of an old pump organ.

When operating Micro-Harmony, the music is entered via the keyboard, one chord at a time, in the following format: N1, N2, N3, N4, T, S. N1 through N4 are the four notes of the chord. The "T" is the time signature for the chord, corresponding to quarter notes, half notes, etc. The "T" may be from .25 to 10. The reference time of "1" may be a whole note, but it doesn't have to be. The "S" is the relative speed, or pace, of the piece and can be anything from 1 on up, with 1 being fastest and 20 (or so) being quite slow. That number is used so that the pace may be changed in the middle of the piece. If the same relative rate is to be maintained, just enter zero for the last part of each chord entry ("S") after defining the rate in the first chord. To repeat all or part of the song, enter "R" for the first note in the chord, and a number in the time signature position to tell how many times to repeat. When executing a repeat, the last entry in the chord tells it which chord to start the repeat on. The chords are numbered by tens, with zero being the first chord, 10 being second, 20 the third, etc.

There is a restriction on the repeat — you may not repeat back to a chord that is later than chord number 250 (the 26th entry). To correct the previous chord entry, enter 100 for the rate, or last position in the chord currently being entered. That signals the program that the chord entry prior to the current one was in error. It will decrement the chord counter by ten and let you enter again. If any (or all) of the notes in a chord are to be rested (not sounded), enter "X" for that note. The time signature will determine the length of the rest.

When playing multi-part music, it's not at all unusual for two or more parts to sound the same note. That's fine for voices or instruments, but bad news for computer-generated music. The reason is that every time a particular note is sounded, its pitch is very precise, and if that note is sounded by two "voices," the two pitches will be identical. However, there's no control over the phase of the two notes relative to each other. Therefore, there is a fifty-fifty chance that the overall sound will diminish in volume, rather than be louder as expected. In fact, it's entirely possible for the two sounds to completely cancel each other — most perplexing.

The answer to the problem is to sound only one of the two or more notes which are the same, and rest the other(s). If two notes are the same, but an octave or more apart, there's no problem. When entering the notes of the chord, a sharp is indicated by a "+" while a flat is shown by "-". Along with the note is a number describing which octave it's in, from 1 to 5. The lowest note (C1) is two octaves below middle C and is the C just below the bass cleff. The note C3 represents middle C. The highest note is B5+ and is the B sharp above the treble staff.

An example of an entered chord might be: C2+,E3,G3,X,.5,3. The first note is C sharp below middle C, the second is E above middle C, the third is G, and the fourth is rested. The time signature is one-half of the reference note, whatever you have chosen that to be, and the relative pace is 3 (fairly fast).

If the music you're entering has more than four parts, normally one or more of the parts may be omitted without hurting too badly. Don't be afraid to experiment. As can be seen in the menu, there is a correction mode. It allows you to place the chord counter anywhere in the piece and begin from there to enter new chords. When finished correcting, press break and type RUN. The chords are placed in memory via pokes, so RUN won't clear them out as would happen if they were in an array. To end the piece, enter a chord of all ones (1, 1, 1, 1, 1, 1). The piece should begin to play almost immediately.

The tones, or notes, are generated by two 8253-integrated circuits (IC's). They consist of three, sixteen-bit counter/dividers each. For the music, we need three from one, and one section from the other. The 8253s are programmed (or "told what to do") by lines 80 to 90 in the music program. We use them as frequency dividers in this application. The pitch of each note is defined by a sixteen-bit number which is written to each section of the 8253 as two bytes of data in the form of low byte first, followed by high byte. The resultant two-byte number is actually a divide ratio of the one-MegaHertz clock.

PLEASE ANSWER THESE 5 QUESTIONS

	YES	NO
1. Do you have information that must be kept organized and accessible?	☐	☐
2. Do you ever need to perform statistical analyses?	☐	☐
3. Would you like to have a tool that will allow you greater flexibility in managing your own or your company's money?	☐	☐
4. Could you use a mail list program that will be easy to use, maintain an unlimited number of names, allow you great sorting flexibility, and even interact with a word processor?	☐	☐
5. Do you own or have access to a TRS-80 microcomputer?	☐	☐

If you answered "yes" to any of these questions, we can be of assistance to you. Our Maxi Series of applications programs are designed to give you maximum versatility with a minimum of hassle. These programs were created for business use, but you'll find yourself using them for personal applications as well.

Thorough support — *Of course, each program comes with in-depth, user-oriented documentation, and is menu-driven to make it easy to use. When necessary, the Maxi programs are compatible with each other, and, whenever pertinent, are interactive with the major word processing and spreadsheet programs published by other manufacturers. Also, we maintain a telephone support line to provide you with any assistance you might require.*

Maxi Manager *by Dale Kubler*

Maxi Manager is a remarkable data base manager. Its fast machine language sort complements its large data storage capacity. The sophistication of its data entry, management, and printing capabilities makes Maxi Manager a versatile tool for many applications.

The program now includes Maxi Utility, which allows you to rescue files on diskettes that have been damaged by excessive wear or misuse and lets you expand, add, or delete fields from an existing data base. Let the unmatched capabilities of Maxi Manager handle your data management!

Model I & Model III. Minimum 1 disk drive required . 012-0196 $149.99

Maxi Mail *by Dale Kubler*

Maxi Mail is a powerful mail list management system that:

1. Is easy to use.
2. Has virtually unlimited storage capacity.
3. Interfaces with the major word processors to generate form letters and other text.
4. Prints mailing labels up to four-across in any format desired.
5. Has virtually unlimited coding capabilities with thirteen fields of information for each record.

Maxi Mail is the most sophisticated user-oriented mail list program available.

TRS-80 Model III only. 2 disk drives required 012-0148 $99.95

Maxi Cras *by Dale Kubler*

(Check Register Accounting System)

Maxi Cras is a system that will computerize check writing, recording and analysis for business and personal finance. The system features 223 income and expense accounts, each of which will handle an unlimited number of transactions. Extensive register and report printing capabilities make Maxi CRAS an indispensible tool for managing money effectively.

Model I & Model III. Minimum 2 drives required 012-0145 $99.95

Maxi Stat *by David Walonick*

Maxi Stat is the most useful statistical analysis package on the market today. It was developed to allow maximum flexibility in designing customized analysis. Maxi Stat handles the three main components of statistical analysis:

1. Complete menu-driven codebook creation and editing.
2. Menu-assisted data entry.
3. User-created control files to describe the statistical analyses to be performed and printed out on the variables of your choice.

Model I & Model III. Minimum 2 drives required 012-0153 $199.95

Send $1.00 for our 16 page booklet "Getting The Most From Your Micro" All 16 pages are packed with indepth explanations and printout samples from the Maxi Series of applications programs.

THE BUSINESS DIVISION
BOX 3435
LONGWOOD,FL 32750
(305) 830-8194

SIMPLY POWERFUL.

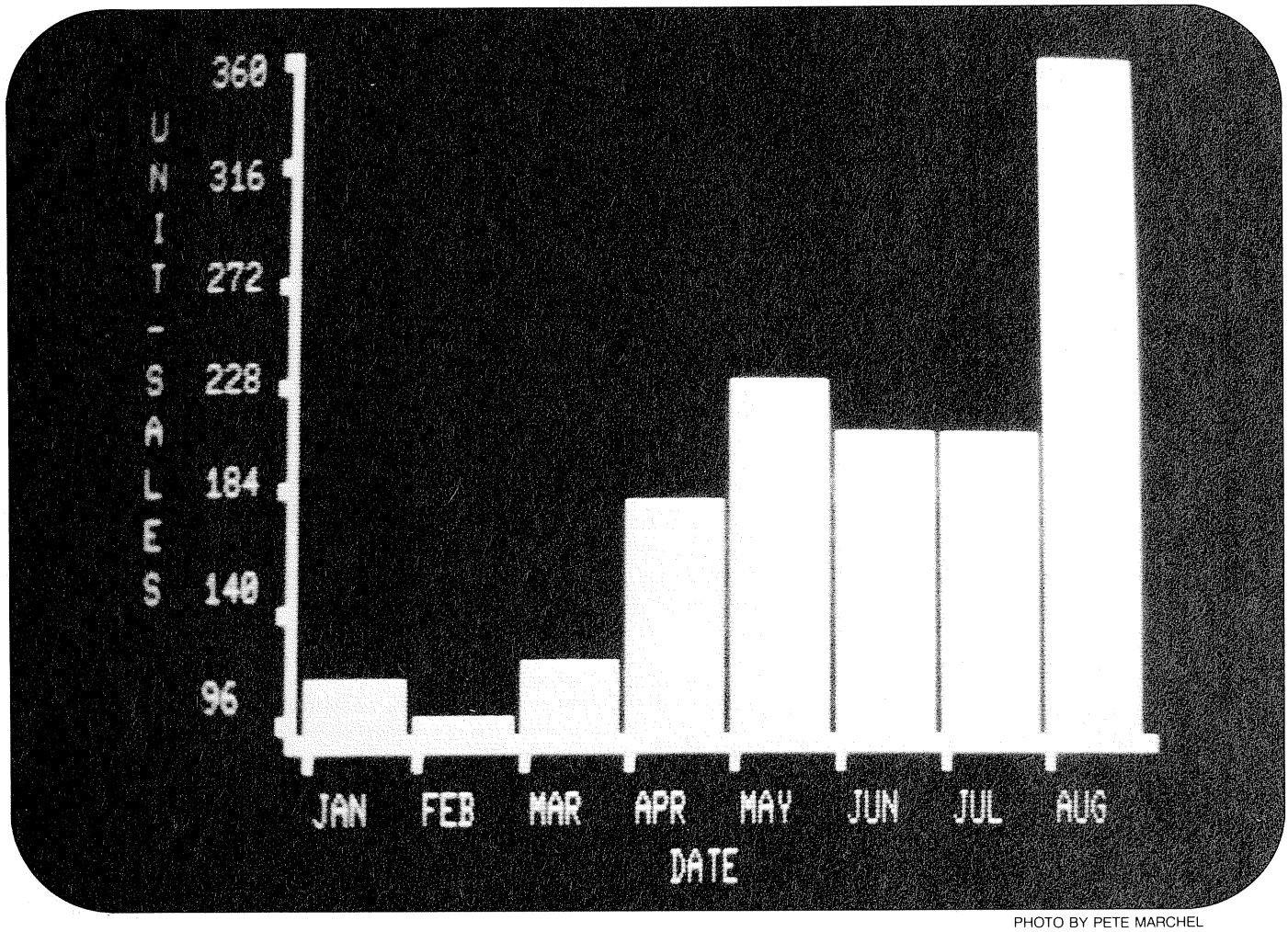


PHOTO BY PETE MARCHEL

SIMPLY POWERFUL

The graph displayed above was created by a simple Lisp program that counts the number of entries it is given and creates a normalized graph with up to 25 bars, their widths adjusted to maximize screen utility. The maximum and minimum entries are found and used to normalize bar height to accentuate these differences so that visual accuracy and screen use are maximized. The vertical labels are rounded to the nearest integer and the marks on the vertical axis are scaled appropriately.

This is only a small step into the wide range of powerful applications software that can be easily created in Lisp.

WHY LISP?

Lisp is the premier language of artificial intelligence because of such features as the similarity of program and data structure and an interactive programming environment. As such, it is capable of application that would be difficult in other computer languages. Applications such as intelligent game programming (like chess), natural language interfaces and symbolic mathematics including integrators, differentiators and algebraic simplifiers are just a few of the areas of AI research.

EVERYDAY APPLICATIONS

The AITech Lisp interpreter was also designed to handle a full range of everyday applications. That's why is has features like graphics and other visual display commands, random access files, while and for loops, double and single precision floating point numbers, multidimensional arrays, trigonometric and exponential functions; features that are not always found on other Lisp interpreters. Of course it comes with a full range of Lisp functions, some of which include function tracing, error trapping, property lists, lambda and nlambda function definitions, strings and string functions. Perhaps most importantly, the AITech Lisp interpreter is one of the fastest higher level language interpreters available for microcomputers on the market today.

THE LISP SYSTEM

The AITech Lisp interpreter for the TRS-80™* is part of a complete Lisp system which includes a Lisp interpreter, an expression oriented Lisp editor and a manual. A symbolic differentiator and algebraic simplifier, a poker player and the bar graph function shown above are also included. The complete system sells for $119.95.

ANTICIPATING NEED

Often, it is the thoughtful details that make programming delightful instead of a chore. That's why the AITech Lisp interpreter has special features like an automatic conversion from integers into floating point numbers, automatic closing of expressions with left and right brackets, fast pretty printing and automatic quoting.

With this much thought put into making a high quality product, it is no wonder we say...

INTELLIGENCE is our middle name

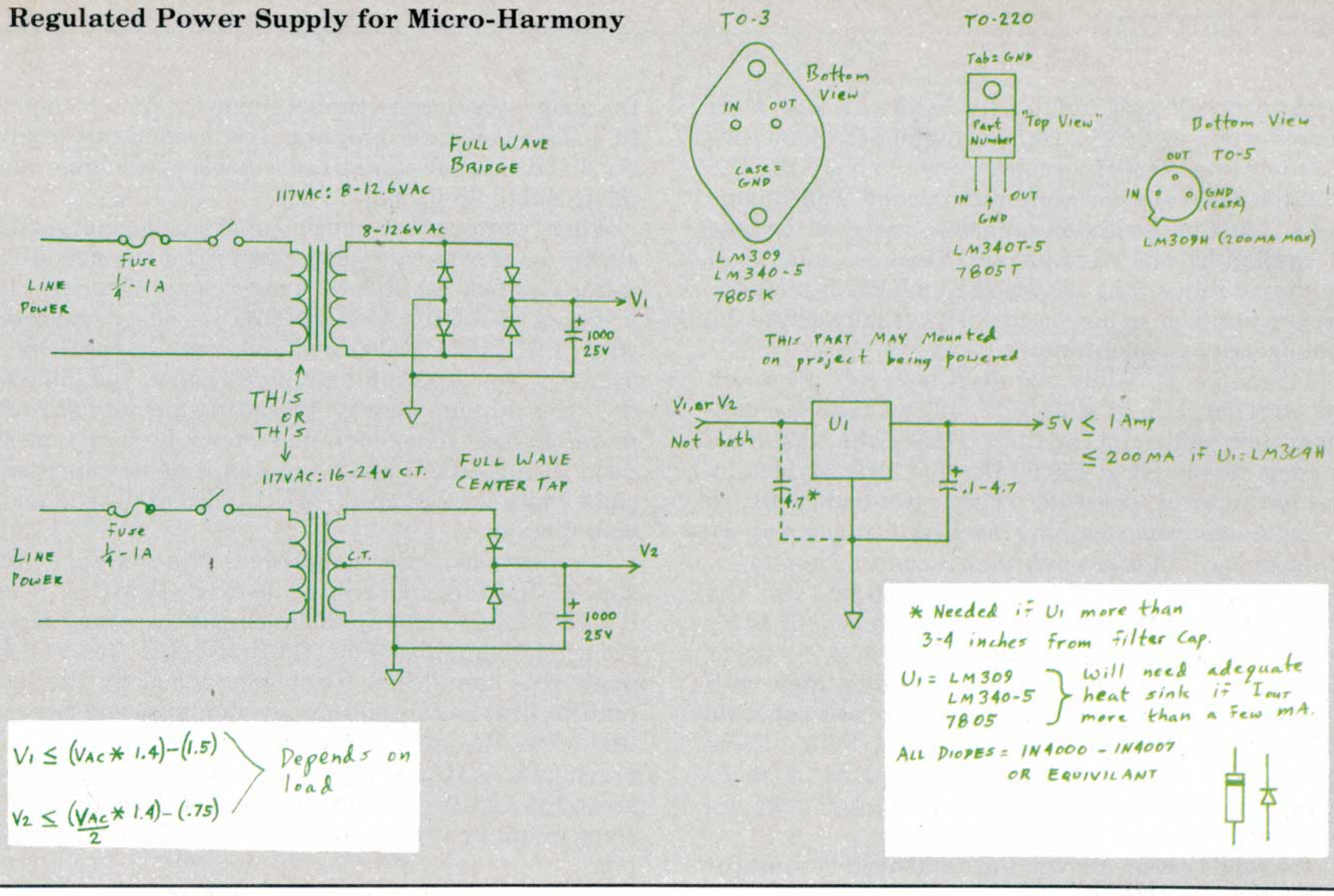
ARTIFICIAL INTELLIGENCE TECHNOLOGIES

For free brochure call **206/644-3068**, or write
Artificial Intelligence Technologies
2121 N.E. 152nd, Redmond, WA 98052

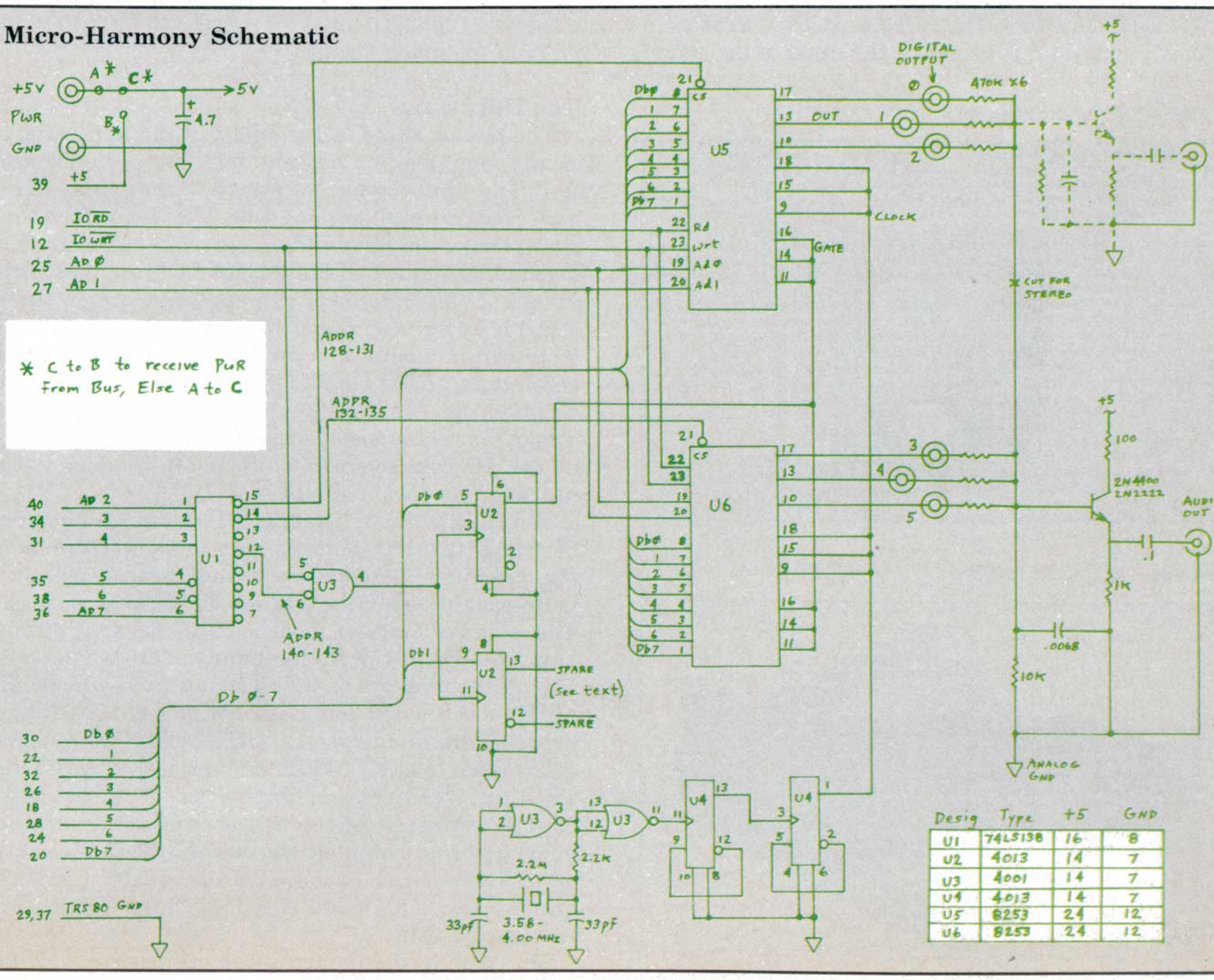
To order please send check or money order, Washington state residents must add sales tax.

*One disk drive required. 48K recommended
TRS-80 is a registered trademark of Tandy Corp

Regulated Power Supply for Micro-Harmony



Micro-Harmony Schematic



Desig	Type	+5	GND
U1	74LS138	16	8
U2	4013	14	7
U3	4001	14	7
U4	4013	14	7
U5	8253	24	12
U6	8253	24	12

Therefore, the lowest note of the system is one MHz divided by 65,536, or 15.2 Hz. The highest frequency we can generate is one MHz divided by two, or 500 KHz. The resolution of the system is one microsecond. That means if we add one to the divide ratio, the period of the note will change by one microsecond. Since we must deal with divide ratios that are whole numbers, or integers, there are limits as to how accurately we may define the pitch (or frequency) of a note.

For example, to define middle A (440 Hz), we would need a divide ratio of 2272.7272, but we may use only integer numbers — no fractions. Hence, the closest we can come to 440 Hz is 439.95 Hz (one MHz divided by 2273) for an error, or 0.012%. That's not bad at all. To prevent a note from sounding, such as during a rest, we just make it so high in pitch that it can't be heard.

Beginning on line 190 of the program, are the definitions of the divide ratios. They are only defined for the lowest octave and are divided by 2, 4, 8, 16, or nothing, depending on the octave definition digit with each note. The ratios are converted to two separate numbers representing low byte/high byte. These numbers are what are actually stored in memory starting at 9C00 hex (40,000 decimal), and later are stored in the 8253 dividers as needed.

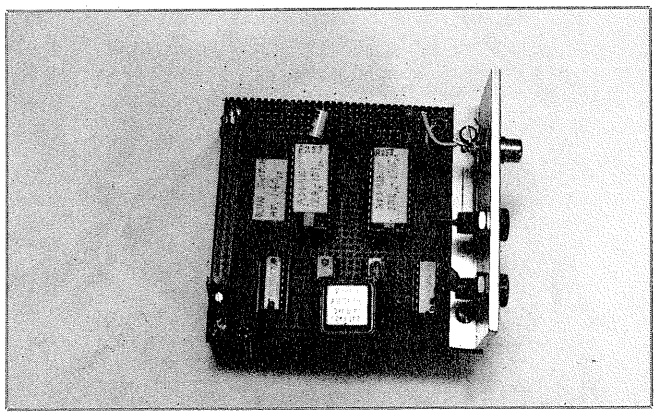
In the tape version shown, the data memory starts at 28,000 decimal. On the schematic, you'll notice a flip/flop (a CMOS 4013 U2). It is used to start and stop the frequency dividers. The only purpose it serves us in the music generator is to silence the beast at the end of the song. It is used in the second program which is the frequency generator program. It is used to restart all of the dividers at the same time to insure predictable phase relationship, if desired.

When running the music program, memory above 40,000 (or 28,000) must be protected for the note divide ratios. In disk BASIC, that may easily be done when bringing up BASIC. Under NEWDOS-80, you must do it with a HIMEM 39999 command while in DOS. Of course, if you are running a tape system, you'll have to do it the old reliable way, by setting memory size after power-up reset. The program is shown for both tape and disk. In the disk version, the song title is truncated to eight characters and has /MUS added at the end for the disk file name.

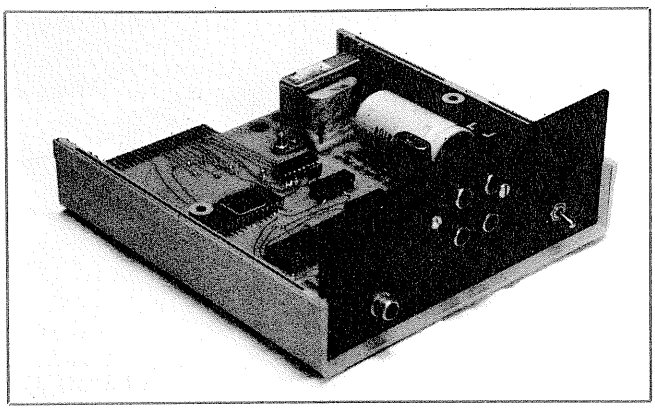
For example, if the song title is "Bach's Fugue in A Minor," the filename will be "BACHSF/MUS." When it comes time to call up the music file, you may type in the whole song title, or just the first eight letters of it, but don't type the /MUS. That gets added by the input routine. Just be careful that you don't use another song title with the first eight characters the same as a previous song. In the previous example, if you entered a second piece titled "Bach's Fugue in C Major," the first song would be lost as the second song overwrites it — pity.

I'm not going to detail the program here. For one thing, it's a bit long for that, and second, I tried to annotate it enough to make it possible to wade through it without too much strain.

Top view — wire wrap version



Side view — PC board version



The Hardware

The photos show two versions of the hardware, one neatly mounted in a box and built on a printed circuit card. The other version shown is a wire wrap model that has served me well for some time. The photo of the boxed model shows only five audio connectors rather than the seven diagrammed on the schematic. I just didn't bring out all of the logic level outputs. They are on the PC board if I ever need them. As may be seen, I bolted on a piece of angle aluminum to mount the power connectors and the audio output connector. The wire-wrap model doesn't have connectors for logic level output for the frequency generator. I just tapped that off of the IC connectors. On the PCB version, I made provision for logic outputs.

Referring to the schematic, U1 serves the function of decoding the I/O address. Since Micro-Harmony is mapped into I/O space, rather than in memory, only the bottom eight bits of the address bus need to be checked. The address may be modified by choosing different outputs of U1. If you do change any addresses, be sure to change the program as well. The run latch, which is half of U2, is set by the command "OUT 140,1." To reset it (and stop the dividers), use "OUT 140,0." The other half of U2 is set by "OUT 140,2" and reset by "OUT 140,0." That latch really has no function as shown, but may be used to control an LED or a beeper for some purpose of your choosing. The latch was there for free, so I hooked it up as shown. If you choose not to connect it, be sure to ground pins 8, 9, 10 and 11 to protect the CMOS device from stray static.

The clock for the dividers is provided by part of U3 and

the four MegaHertz crystal. You can substitute a 3.58 MHz color burst crystal, if desired. To maintain the absolute accuracy of the music generated, change the divider constants in the program starting at line 230. Multiply each number by 0.895 to make the notes true.

The clock is divided down to 1 MHz (or 0.895 MHz) by the two sections of U4. U5 and U6 are the two 8253s used as frequency dividers. Their outputs are attenuated by the 470K and 10K resistors, with filtering being provided by the 0.0068 uF capacitor across the 10K resistor. The transistor is an emittor follower used to buffer the output to the outside world. Lest anyone should wonder, the 100-Ohm resistor in the collector of the transistor is used to protect it against shorts in the output. If stereo output is desired, just wire up a second output stage as shown in the dotted portion of the schematic.

There's nothing fussy about the layout. The only possible problem area is the grounding for the output stage. It would be best to return it directly to power ground, along with the ground for the input network connected to the base. For the connection to the TRS-80 bus, I used a 40-pin board mount connector which mates with a crimp-on ribbon cable connector. The other end of the ribbon cable plugs into either the keyboard expansion port (in place of the expansion interface) or into the bus expansion port of the expansion interface, if one is connected. Micro-Harmony requires five volts at 300MA to run, and do be careful of the polarity when connecting it up. Measure any voltage before applying it to your music generator for the first time. You can see in the photo of the wire wrapped model that I used a big, clunky, 4-MHz crystal. There wasn't room for it on the board, so I glued it to the tops of U3 and U4. I admit it gladly: I'm a cheapskate. Hopefully, the unit you use will fit better. If I have to change one of those chips, I will cry a lot!

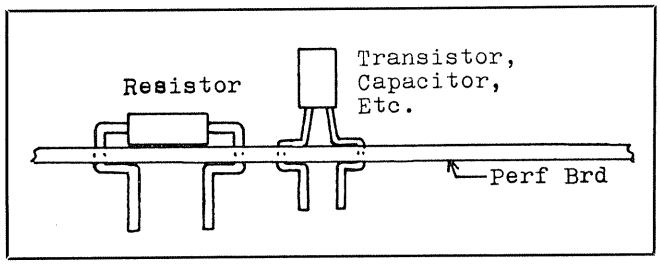
After taking care of the mechanical matters (namely gluing in place the IC connectors, making sure that all of the number one pins go the same way) and bolting on the angle aluminum (if any), etc., go ahead and begin hooking up the power to the IC sockets. Use 28-gauge (or larger) wire for this part to keep impedances low. I did mine with the original 28-gauge slit-and-wrap. If you do the same, be sure that the wire doesn't rub against any other pins or screws. The coating is fragile and will allow a short circuit, often long after the device is first turned on.

I wired this whole unit with the stuff, but I definitely don't recommend it. Wire wrap is a bit slower to use, but much better in the long run. If you use normal hookup wire for the power, solder it to the appropriate pins on the sockets near the perf board. That will allow room for later wire wrapping on the same pins, if necessary.

After hooking up the power leads, wire in the forty-pin bus, whether it be with the connector or the ribbon cable directly. Incidentally, if you do forgo the connector at this end, be sure to glue the ribbon cable down to the perf

board to avoid broken wires later. Double check the bus connections as mistakes here can cause strange problems.

Mounting Parts Onto PC Board



I installed an extra 16-pin socket for the output resistors (six each, 470K, and one 10K) and the 0.0068 uF capacitor. It allows me to easily change the filtering, output levels, and such. It also provides for easy connection to the small components. For the parts that you install directly to the perf board, just run the part's lead through the board, "clinch" it on the other side, then bend it back away from the board. You wind up with two 90-degree bends with a short length of the lead flat against the board holding the part tightly (see the figure). To make connection, just wire wrap to the lead, then follow up with a bit of solder. As you wire the unit, it's a good idea to mark off each wire on the schematic as you hook it up. It's a great help to avoid errors and omissions.

When you're all done, connect power to the unit *without IC's* and check power at each IC socket. Also, note the polarity of the voltage and make sure it is correct. If all is okay, remove power and install the ICs. Double and triple check that they are in the right way, as it will probably destroy any that are backward. Apply power and repeat the check for voltage on each IC. Also, check power ground on each IC for zero volts (a few millivolts is okay). The connections for power and ground for each IC type are given in a table on the schematic. If you're okay so far, we're ready for the big test.

Connect the ribbon cable from the music generator to the proper connector on your TRS-80 (which is also turned off). Apply power to Micro-Harmony first, then turn on your system. If the screen comes on full of garbage and reset won't clear it, you probably plugged in the bus connector backward. No harm done, so calm down and turn the connector around.

Now, enter one of the programs and make it sound pretty. At this point, you'll need to connect the output of Micro-Harmony to one of the following devices: Auxiliary or tuner input on an audio amplifier, or auxiliary input of a tape recorder, or to a pair of headphones. The last choice may be made necessary by other people in the house that don't appreciate budding

musical genius. Incidentally, it's normal for the 8253s to run a bit warm, but nothing should get uncomfortably hot.

If it doesn't work, go back and recheck the wiring, especially the 40-pin TRS-80 bus. Also, check that the crystal is oscillating. A logic analyzer applied to U4 pin 1 should indicate pulses. If a logical analyzer isn't available, I suggest you consider getting one or building one. Even a cheapy is worth its weight in gasoline. However, if all that you have is a voltmeter, measure pin 1 of U4 on the 5-volt DC scale. Since the clock signal at that point has a 50-50 duty cycle, the meter should read around 2.5 volts.

If the voltage reads either zero or +5, move your test lead to U3 pin 11. If it reads around +2.5v there, either U4 is bad, or it isn't wired correctly. If there doesn't seem to be any activity on U3 pin 11, decrease the 33 pF caps to 10 pF or so. If it still won't go, change U3, then the crystal, in that order. This assumes that you rechecked your wiring first. Don't be bashful. We all make mistakes.

If this device looks like something you'd like to build, but you'd rather not mess with the wiring, take heart. I will provide a two-sided, bare, printed circuit card for $35, or a full kit of parts (including the PC board) for $85. If you can't even face soldering the parts into the board, I'll do it for you for $120 total, assembled and tested. A three-foot ribbon cable with a plug on each end to plug into your TRS-80 is $25 (for PC version, only). In addition, I'll add cable and connectors in one-foot increments for $10 each. The idea is that you can plug in several of your peripherals with the one cable. Don't ask for more than four extra connectors (in addition to the two connectors already provided). It's likely your system can't drive that much cable without external buffering.

Ed note: You may contact Mr. Bennett at 15853 7th S.W., Seattle, WA 98166.

Parts List

IC's

74LS138	U1
4013 (CMOS)	U2, U4
4001 CMOS	U3
8253 (NMOS)	U5, U6

Resistors

2.2 Meg, 5%, ¼W	1 each
470 K, 5%, ¼W	6 each
27 K, 5%, ¼W	1 each
10 K, 5%, ¼W,	1 each
2.2 K, 5%, ¼W	1 each
1 K, 5%, ¼W	1 each
100 Ohm, 5%, ¼W	1 each

Capacitors

4.7 uF, 25v-50v, Electrolytic ..... 1 each
0.0068 uF, Disc Ceramic ..... 1 each
0.1 uF, Disc Ceramic or Monolythic ..... 3 each
33 pF, Disc Ceramic ..... 2 each

Miscellaneous

Quartz Crystal, 4 MHz or
3.579545 MHz color burst ..... 1 only
IC Sockets, wire-wrap pins
14-pin ..... 3 each
16-pin ..... 2 each
24-pin ..... 2 each
2N2222A or 2N4400 Transistor, NPN ..... 1 each
Jacks, RCA Phono (Audio)..... As Needed
Jacks, Banana type or personal preference
(For power input) ..... As Needed
Perf Board, 0.064" thick x 4" x 4" (minimum with 0.1" spaced hole pattern (Vector, or equivalent)

MUSDISK/BAS — Disk Version, Model I, 32K

```
1 ' THIS IS THE VERSION FOR MOD I, VER I
I, 32K, DISK
5 '**** MICRO-HARMONY BY RAY BENNETT
  DEC 78 ****
10 CLS:CLEAR500
15 MM=(&H9C00)
20 PRINT"TO ENTER MUSIC, ENTER FORMAT C3
+(MIDDLE C SHARP)"
30 PRINT"ENTER BAS,TENOR,ALTO,SOPRANO,TI
ME(IN EIGHTH NOTES),"
40 PRINT"TEMPO (0 IF NO CHANGE, 1 IS FAS
TEST, 35 IS QUITE SLOW)."
50 PRINT"X FOR THE NOTE = REST.  100 FOR
 TEMPO WILL RETURN TO"
60 PRINT"PREVIOUS LINE. R FOR NOTE MEANS
 REPEAT TO BEGINNING."
70 PRINT"WHEN REPEATING, TIME SIGNATURE=
NUMBER OF REPEATS."
80 OUT131,54:OUT131,118:OUT131,182
90 OUT135,54:OUT135,118:OUT135,182
100 GOTO780
110 I=0
120 INPUT B$(0),B$(2),B$(4),B$(6),T,S:PR
INT "LINE NUMBER ",I
130 IFB$(0)<>"R"THEN T=T*4
140 IFB$(0)<>"R"ANDS>99THENPRINT"REPEAT
LINE #";I-10:I=I-10:GOTO120
150 FOR J=0TO6 STEP2
160 C$=LEFT$(B$(J),1):' NOTE IN C$
```

```
17Ø D$=MID$(B$(J),2,1):D=VAL(D$):'OCTAVE
 IN D
18Ø IF LEN(B$(J))>2 THEN E$=RIGHT$(B$(J)
,1) ELSE E$=""
19Ø '** THE FOLLOWING IS THE TABLE LOOK-
UP FOR DIVIDE RATIOS
2ØØ IF C$="R" THEN C=3ØØØØ
21Ø C$=C$+E$
22Ø IF C$="X" THEN C=2
23Ø IF C$="C" THEN C=15288
24Ø IF C$="C+"OR C$="D-" THEN C=1443Ø
25Ø IF C$="D" THEN C=1362Ø
26Ø IF C$="D+" OR C$="E-" THEN C=12855
27Ø IF C$="E" THEN C=12134
275 IF C$="E+" THEN C=11453
28Ø IF C$="F" THEN C=11453
29Ø IF C$="F+" OR C$="G-" THEN C=1Ø811
3ØØ IF C$="G" THEN C=1Ø2Ø4
31Ø IF C$="G+" OR C$="A-" THEN C=9632
32Ø IF C$="A" THEN C=9Ø91
33Ø IF C$="A+" OR C$="B-" THEN C=8581
34Ø IFC$="B" THEN C=8Ø99
35Ø '** FOLLOWING DETERMINES PROPER OCTA
VE.
36Ø IF D=2 THEN C=INT(C/2)
37Ø IF D=3 THEN C=INT(C/4)
38Ø IF D=4 THEN C=INT(C/8)
39Ø IF D=5 THEN C=INT(C/16)
4ØØ IF D>5 THEN PRINT"IMPROPER OCTAVE":G
OTO12Ø
41Ø GOSUB 127Ø
42Ø PRINT C;
43Ø NEXT J
44Ø PRINT
45Ø POKE (MM+8+I),T
46Ø POKE(MM+9+I),S
47Ø IF T=Ø THEN 51Ø
48Ø I=I+1Ø
49Ø GOTO12Ø
5ØØ PRINT N$:'** THE FOLLOWING PLAYS THE
 MUSIC
51Ø I=Ø
52Ø A1=PEEK(MM+I):A2=PEEK(MM+1+I)
53Ø B1=PEEK(MM+2+I):B2=PEEK(MM+3+I)
54Ø C1=PEEK(MM+4+I):C2=PEEK(MM+5+I)
55Ø D1=PEEK(MM+6+I):D2=PEEK(MM+7+I)
56Ø T=PEEK(MM+8+I)
565 S=PEEK(MM+9+I)
57Ø IFT=ØTHENOUT14Ø,Ø:GOTO78Ø
58Ø IFT4=1THEN66Ø
```

```
590 IFA1=48ANDA2=117THENT4=1:PRINT"REPEA
T TO";S:T3=T:I=S:GOTO520
600 IFS>ØANDT4=ØTHENT1=S+T2
610 OUT128,A1:OUT128,A2:OUT129,B1:OUT129
,B2
620 OUT130,C1:OUT130,C2:OUT132,D1:OUT132
,D2
630 OUT140,1:FORJ=1TOT1:NEXTJ
640 T=T-1:IFT>ØTHEN630
650 I=I+10:GOTO520
660 IFA1=48ANDA2=117THENT3=T3-1:IFT3>ØTH
ENI=S:GOTO520
670 IF T3>Ø THEN 610
680 T4=Ø:GOTO650
690 '** THE FOLLOWING LISTS THE MUSIC SH
OWING DIVIDE RATIOS
700 I=Ø
710 PRINT I;PEEK(MM+I);PEEK(MM+1+I);PEEK
(MM+2+I);
720 PRINT PEEK(MM+3+I);PEEK(MM+4+I);PEEK
(MM+5+I);PEEK(MM+6+I);PEEK(MM+7+I);PEEK(
MM+8+I);PEEK(MM+9+I)
730 IF PEEK(MM+8+I)=Ø THEN 760
740 I=I+10
750 GOTO 710
760 PRINT@960,"ANY KEY TO CONTINUE."
770 IF INKEY$="" THEN 770ELSE GOTO 780
780 PRINT:PRINT"**** FUNCTIONAL MENU ***
*"
790 PRINT"TO ENTER NEW SONG,          TY
PE 1
800 PRINT"TO REPLAY SONG,             TY
PE 2
810 PRINT"TO CHANGE TEMPO,            TY
PE 3
820 PRINT"TO CORRECT ENTERED MUSIC,   TY
PE 4
830 PRINT"TO SAVE MUSIC DATA ON DISK, TY
PE 5"
840 PRINT"TO RETRIEVE DATA FROM MEDIA,TY
PE 6"
850 PRINT"TO SEE DATA TABLE,          TY
PE 7"
860 PRINT"TO START IN MIDDLE OF SONG, TY
PE 8"
865 INPUT M
880 ON M GOTO 890,500,910,960,980,1110,7
00,2000
885 PRINT"ILLEGAL ENTRY.":GOTO780
890 INPUT"WHAT IS THE NAME OF THE SONG T
O BE ENTERED";N$
900 GOTO110
910 PRINT"PRESENT TEMPO IS ";PEEK(MM+9)+
T2
920 INPUT"ENTER TEMPO MODIFIER (+OR- NUM
BER TO ADD TO TEMPO)";T2
930 T1=PEEK(MM+9)+T2
```

```
935 IF T1<=0 THEN T1=1:PRINT"TEMPO CAN'T
 BE FASTER THAN 1"
940 PRINT"CURRENT TEMPO NUMBER IS NOW";T
1
950 GOTO 510
960 INPUT"ENTER STARTING LINE NUMBER";I
970 GOTO120
980 IF N$<>"" PRINT "DUMP TO DISK OF -
  ";N$
990 IF N$=""INPUT"DUMP TO DISK. WHAT IS
THE SONG TITLE";N$
995 K$=""
1000 OPEN"O",1,LEFT$(N$,8)+"/MUS"
1020 PRINT I;"NOTES"
1030 PRINT#1,N$
1035 PRINT#1,I
1040 FOR J=1TOI+10
1050 K=PEEK(MM+J-1):K$=K$+STR$(K)
1060 IF J/50=INT(J/50) THEN PRINT#1,K$:K
$=""
1070 NEXT J
1080 PRINT#1,K$:K$=""
1090 PRINT#1,"END"
1095 CLOSE
1100 PRINT"ALL DONE WITH DUMP.":GOTO780
1110 INPUT"LOAD FROM TAPE OR DISK (T/D)"
;LX$:IF LX$="D" THEN 5000
1115 J1=0:PRINT"LOAD FROM TAPE. READY TA
PE RECORDER. ANY KEY."
1120 IF INKEY$="" THEN 1120
1125 CMD"T"
1130 INPUT#-1,N$,I:PRINT "TITLE IS -
";N$
1140 PRINT I
1150 I1=I
1160 INPUT#-1,K$
1170 IF K$="END" PRINT"DATA LOADED":GOTO
780
1180 J2=LEN(K$)
1190 FOR J=1 TO J2
1200 IF MID$(K$,J,1)=" " POKE MM+J1,K2:J
1=J1+1:K2$="":NEXT J:GOTO1250
1220 K2$=K2$+MID$(K$,J,1)
1230 K2=VAL(K2$)
1240 NEXT J
1250 POKE MM+J1,K2:J1=J1+1:K2$=""
1260 GOTO1160
1270 'SUBROUTINE TO CONVERT DIVISORS TO
TWO BYTES EACH
1275 IF C>255 THEN C2=INT(C/256)
1280 IF C>255 THEN C1=C-(C2*256) ELSE C1
=C
1290 IF C=2 THEN C1=2:C2=0
1300 POKE(MM+I+J),C1:POKE(MM+1+I+J),C2
1310 RETURN
2000 INPUT"WHAT LINE DO YOU WANT TO BEGI
N AT";I
```

```
2010 INPUT"WHAT IS THE TEMPO";T1
2030 GOTO520
5000 J1=0:INPUT"LOAD FROM DISK. WHAT IS
THE SONG NAME";N$
5010 OPEN"I",1,LEFT$(N$,8)+"/MUS"
5020 INPUT#1,N$:PRINT "TITLE IS -   ";N$
5025 INPUT#1,I
5030 PRINT I;" CHORD ENTRIES
5040 I1=I
5050 INPUT#1,K$
5060 IF K$="END" PRINT"DATA LOADED":CLOS
E:GOTO780
5070 J2=LEN(K$)
5080 FOR J=1 TO J2
5090 IF MID$(K$,J,1)=" " POKE MM+J1,K2:J
1=J1+1:K2$="":NEXT J:GOTO1250
5100 K2$=K2$+MID$(K$,J,1)
5110 K2=VAL(K2$)
5120 NEXT J
5130 POKE MM+J1,K2:J1=J1+1:K2$=""
5140 GOTO505Ø
```

MUSTAPE/BAS — Tape Version, Model I, 16K

```
1 'THIS IS THE VERSION FOR A MODEL I, LE
VEL II, NO DISK
5 '****  MICRO-HARMONY BY RAY BENNETT  *
***
10 CLS:CLEAR1000
20 PRINT "TO ENTER MUSIC, ENTER FORMAT C
3+ (= MIDDLE C SHARP)"
30 PRINT"ENTER BAS,TENOR,ALTO,SOPRANO,TI
ME(IN EIGHTH NOTES),"
40 PRINT"TEMPO (0 IF NO CHANGE, 1 IS FAS
TEST, 35 IS QUITE SLOW)."
50 PRINT"X FOR THE NOTE = REST.  100 FOR
 TEMPO WILL RETURN TO"
60 PRINT"PREVIOUS LINE. R FOR NOTE MEANS
 REPEAT TO BEGINNING."
70 PRINT"WHEN REPEATING, TIME SIGNATURE=
NUMBER OF REPEATS."
80 OUT131,54:OUT131,118:OUT131,182:'CONF
IG 8253'S

90 OUT135,54:OUT135,118:OUT135,182
100 GOTO780
105 'ENTER CHORDS FROM KYBRD
110 I=0

120 INPUT B$(0),B$(2),B$(4),B$(6),T,S:PR
INT "LINE NUMBER ",I
130 IFB$(0)<>"R"THEN T=T*4
140 IFB$(0)<>"R"ANDS>99THENPRINT"REPEAT
```

```
LINE #";I-1Ø:I=I-1Ø:GOTO12Ø
15Ø FOR J=ØTO6 STEP2
16Ø C$=LEFT$(B$(J),1):' NOTE IN C$
17Ø D$=MID$(B$(J),2,1):D=VAL(D$):'OCTAVE
 IN D
18Ø IF LEN(B$(J))>2 THEN E$=RIGHT$(B$(J)
,1) ELSE E$=""
19Ø '** THE FOLLOWING IS THE TABLE LOOK-
UP FOR DIVIDE RATIOS
2ØØ IF C$="R" THEN C=3ØØØØ:'CODE FOR REP
EAT
21Ø C$=C$+E$
22Ø IF C$="X" THEN C=2
23Ø IF C$="C" THEN C=15288
24Ø IF C$="C+"OR C$="D-" THEN C=1443Ø
25Ø IF C$="D" THEN C=1362Ø
26Ø IF C$="D+" OR C$="E-" THEN C=12855
27Ø IF C$="E" THEN C=12134
275 IF C$="E+" THEN C=11453
28Ø IF C$="F" THEN C=11453
29Ø IF C$="F+" OR C$="G-" THEN C=1Ø811
3ØØ IF C$="G" THEN C=1Ø2Ø4
31Ø IF C$="G+" OR C$="A-" THEN C=9632
32Ø IF C$="A" THEN C=9Ø91:'11Ø HZ
33Ø IF C$="A+" OR C$="B-" THEN C=8581
34Ø IF C$="B" THEN C=8Ø99
35Ø '** FOLLOWING DETERMINES PROPER OCTA
VE.
36Ø IF D=2 THEN C=INT(C/2)
37Ø IF D=3 THEN C=INT(C/4)
38Ø IF D=4 THEN C=INT(C/8)
39Ø IF D=5 THEN C=INT(C/16)
4ØØ IF D>5 THEN PRINT"IMPROPER OCTAVE":G
OTO12Ø
41Ø GOSUB 127Ø:'CONVERT DIVIDER TO 2 BYT
ES & STORE
42Ø PRINT C;
43Ø NEXT J
44Ø PRINT
45Ø POKE (28ØØ8+I),T:'STORE TIME SIGNATU
RE
46Ø POKE(28ØØ9+I),S:'STORE RATE
47Ø IF T=Ø THEN 51Ø
48Ø I=I+1Ø
49Ø GOTO12Ø
5ØØ PRINT N$:'** THE FOLLOWING PLAYS THE
 MUSIC
51Ø I=Ø
515 'RETRIEVE DATA FROM MEMORY
52Ø A1=PEEK(28ØØØ+I):A2=PEEK(28ØØ1+I)
53Ø B1=PEEK(28ØØ2+I):B2=PEEK(28ØØ3+I)
54Ø C1=PEEK(28ØØ4+I):C2=PEEK(28ØØ5+I)
```

```
550 D1=PEEK(28006+I):D2=PEEK(28007+I)
560 T=PEEK(28008+I)
565 S=PEEK(28009+I)
570 IFT=0THENOUT140,0:GOTO780
580 IFT4=1THEN660
590 IFA1=48ANDA2=117THENT4=1:PRINT"REPEA
T TO";S:T3=T:I=S:GOTO520
600 IFS>0ANDT4=0THENT1=S+T2
605 'NOW STORE IN 8253 - ONE CHORD
610 OUT128,A1:OUT128,A2:OUT129,B1:OUT129
,B2
620 OUT130,C1:OUT130,C2:OUT132,D1:OUT132
,D2
630 OUT140,1:FORJ=1TOT1:NEXTJ
640 T=T-1:IFT>0THEN630
650 I=I+10:GOTO520
660 IFA1=48ANDA2=117THENT3=T3-1:IFT3>0TH
ENI=S:GOTO520
670 IF T3>0 THEN 610
680 T4=0:GOTO650
690 '** THE FOLLOWING LISTS THE MUSIC SH
OWING DIVIDE RATIOS
700 I=0
710 PRINT I;PEEK(28000+I);PEEK(28001+I);
PEEK(28002+I);
720 PRINT PEEK(28003+I);PEEK(28004+I);PE
EK(28005+I);PEEK(28006+I);PEEK(28007+I);
PEEK(28008+I);PEEK(28009+I)
730 IF PEEK(28008+I)=0 THEN 760
740 I=I+10
750 GOTO 710
760 PRINT@960,"ANY KEY TO CONTINUE."
770 IF INKEY$="" THEN 770ELSE GOTO 780
780 PRINT:PRINT"**** FUNCTIONAL MENU ***
*"
790 PRINT"TO ENTER NEW SONG,          TY
PE 1
800 PRINT"TO REPLAY SONG,             TY
PE 2
810 PRINT"TO CHANGE TEMPO,            TY
PE 3
820 PRINT"TO CORRECT ENTERED MUSIC,   TY
PE 4
830 PRINT"TO SAVE MUSIC DATA ON TAPE, TY
PE 5"
840 PRINT"TO RETRIEVE DATA FROM TAPE, TY
PE 6"
850 PRINT"TO SEE DATA TABLE,          TYP
E 7"
860 PRINT"TO START IN MIDDLE OF SONG,  T
YPE 8"
865 INPUT M
870 IF M>8 PRINT"ILLEGAL ENTRY.":GOTO780
880 ON M GOTO 890,500,910,960,980,1110,7
00,2000
890 INPUT"WHAT IS THE NAME OF THE SONG T
O BE ENTERED";N$
900 GOTO110
910 PRINT"PRESENT TEMPO IS ";PEEK(28009)
+T2
920 INPUT"ENTER TEMPO MODIFIER (+OR- NUM
BER TO ADD TO TEMPO)";T2
930 T1=PEEK(28009)+T2
935 IF T1<=0 THEN T1=1:PRINT"TEMPO CAN'T
 BE FASTER THAN 1"
940 PRINT"CURRENT TEMPO NUMBER IS NOW";T
1
950 GOTO 510
960 INPUT"ENTER STARTING LINE NUMBER";I
970 GOTO120
980 IF N$<>"" PRINT "DUMP TO TAPE OF -
  ";N$
990 IF N$=""INPUT"DUMP TO TAPE. WHAT IS
THE SONG TITLE";N$
1000 PRINT"READY RECORDER. ANY KEY TO RE
CORD."
1010 IF INKEY$="" THEN 1010
1020 PRINT I;"NOTES"
1030 PRINT#-1,N$,I
1040 FOR J=1TOI+10
1050 K=PEEK(28000+J-1):K$=K$+STR$(K)
1060 IF J/50=INT(J/50) THEN PRINT#-1,K$:
K$=""
1070 NEXT J
1080 PRINT#-1,K$:K$=""
1090 PRINT#-1,"END"
1100 PRINT"ALL DONE WITH DUMP.":GOTO780
1110 J1=0:PRINT"LOAD FROM TAPE. READY TA
PE RECORDER. ANY KEY."
1120 IF INKEY$="" THEN 1120
1130 INPUT#-1,N$,I:PRINT "TITLE IS -
";N$
1140 PRINT I
1150 I1=I
1160 INPUT#-1,K$
1170 IF K$="END" PRINT"DATA LOADED":GOTO
780
1180 J2=LEN(K$)
1190 FOR J=1 TO J2
1200 IF MID$(K$,J,1)=" " POKE28000+J1,K2
:J1=J1+1:K2$="":NEXT J:GOTO1250
1220 K2$=K2$+MID$(K$,J,1)
1230 K2=VAL(K2$)
1240 NEXT J
1250 POKE 28000+J1,K2:J1=J1+1:K2$=""
1260 GOTO1160
1270 'SUBROUTINE TO CONVERT DIVISORS TO
TWO BYTES EACH
1275 IF C>255 THEN C2=INT(C/256)
1280 IF C>255 THEN C1=C-(C2*256) ELSE C1
=C
1290 IF C=2 THEN C1=2:C2=0
1300 POKE(28000+I+J),C1:POKE(28001+I+J),
C2
```

```
1310 RETURN
2000 INPUT"WHAT LINE DO YOU WANT TO BEGI
N AT";I
2010 INPUT"WHAT IS THE TEMPO";T1
2030 GOTO520
```

FREQGEN/BAS — Frequency Generator

```
10 '***  FREQUENCY GENERATOR  ***
20 '       BY RAY BENNETT
30 '
40 CLS
45 OUT131,54:OUT131,118:OUT131,182:'CONF
IGURE 8253'S
46 OUT135,54:OUT135,118:OUT135,182
50 INPUT"HOW MANY CHANNELS (1-6)";C
60 IF C<1 OR C>6 THEN 50
70 INPUT"FREQ CHANGE IN PHASE (Y/N)";F$:
F$=LEFT$(F$,1)
80 FOR I=1TOC
90 PRINT"INPUT FREQUENCY FOR CHAN ";I
100 INPUT F(I)
110 GOSUB 1000:'DETERMINE CLOSEST FREQ
120 OUT 127+I,D1:OUT127+I,D2:'DATA TO 82
53
130 NEXT I
140 OUT140,1
150 GOTO500
500 IF C=1 THEN C1=1:GOTO520
505 PRINT"INPUT CHAN # TO BE CHANGED (1
TO ";C;")    ";
510 INPUT C1:IF C1<1 OR C1>C GOTO500
515 I=C1
520 INPUT"NEW FREQUENCY";F(I)
525 IF F(I)=0 THEN OUT140,0:GOTO500
530 GOSUB1000
540 OUT127+C1,D1:OUT127+C1,D2
550 IF F$="Y" THEN OUT140,0:OUT140,1:'ST
ART IN PHASE
560 GOTO 500
1000 'CONVERT DATA TO TWO BYTES
1001 IF F(I)>250000 PRINT"FREQ TOO HIGH"
:F(I)=250000
1002 IF F(I)<15.7 PRINT"FREQ TOO LOW":F(
I)=15.7
1003 F(I)=1000000/F(I):X=F(I)-INT(F(I))
1005 IF X>=.5 THEN F(I)=INT(F(I))+1 ELSE
 F(I)=INT(F(I))
1007 PRINT"CLOSEST ACTUAL FREQUENCY=";(1
000000/F(I))
1010 IF F(I)>255 THEN D2=INT(F(I)/256) E
LSE D2=0
1020 IF F(I)>255 THEN D1=F(I)-(D2*256) E
LSE D1=F(I)
1030 RETURN
```

Exploring VisiCalc

Models I/II/III

Timothy K. Bowman, Contributing editor

Before we dive into this month's topic, I would like to share with you some first impressions of my new Radio Shack Model 4 computer and the impact it will have on VisiCalc users. By now, most of you have heard of the new Model 4. I have had one for a short time and am quite impressed. Besides being packaged in an attractive white case, it runs all of the Model III software, plus, it will run CP/M software at a 5-megahertz clock speed. With CP/M, all of the other CP/M-based electronic spreadsheet programs will be at your fingertip (or should I say pocketbook?). The ability to run both Model III and CP/M gives the Model 4 user the best of two worlds: Radio Shack and virtually all of the other eight-bit software world.

If you are considering purchasing a Model 4, be aware of the following points. First, the Model 4 is entirely a RAM-based machine. That is to say, Model III ROM calls are not allowed. That RAM base, however, produces a very flexible machine because you could design your own operating system. Second, if you do decide to purchase one, be sure to purchase the technical manual as none of the absolute RAM addresses are included with the machine and you need those addresses to do any PEEKing and POKEing (as in video display of Model III alternate characters while operating in the Model 4 mode).

Third, while operating in the Model 4 mode, cassette input or output is not supported except for creating Model 100 (portable computer) tapes. Fourth and last, it will be a while before the TRSDOS 6.0-based software is available to run in the Model 4 mode. Keep your eye on *Basic Computing* for conversion programs to convert Model III BASIC to TRSDOS 6.0 Microsoft BASIC(3). Watch this column for possible ways to run Model III VisiCalc using Model 4 TRSDOS 6.0. In the long run, with its speed, 128K memory size, and a hard disk, I believe the Model 4 will become the workhorse for Tandy and surpass the Model 12. The VisiCalc user will be able to create larger spreadsheets and have them recalculated quickly. So much for the editorial, let's turn to our topic this month, using Scripsit and VisiCalc.

Why Bother?

Some of you may be asking why anyone would want to use Scripsit and VisiCalc. After all, one is a word processor and the other crunches numbers. That, my friends, is precisely the reason they can be used to complement one another. After creating a VisiCalc file and saving it to disk, try loading it into Scripsit. Yes, it will load. Try it.

As a test, in VisiCalc, type in the short program shown in Figure 1. The program is nonsensical, but it will serve our purpose. Save it to disk under the name JULYTEST (the VisiCalc program will add the suffix VC and if you want to specify a drive number, be sure to do that). Exit VisiCalc and load Scripsit.

When you have loaded Scripsit, get into the Command mode by pressing the BREAK key. At the prompt, SPECIAL COMMAND?, type L JULYTEST/VC. When the load is complete, you should see a screen display much like Figure 2. Let's look a little closer at that screen.

Screen Contents

The first eight lines show the actual cell contents of our little file. Each line shows the cell location, then a colon, any format instructions, and the contents (either a label or a value). The last six lines give us the following information: The display is one window (/W1); the order of calculation is by column (/GOC); the recalculation mode is automatic (/GRA); the global format is integer (/GFI); the global column width setting is nine characters wide (/GC9); and the cursor position is at A1. Wasn't that simple? You probably are now even remembering that those are the same commands that you used while you were operating VisiCalc.

Figure 1

```
This is the cell content of ARTCL783/VC:1:

        A          B          C          D          E

1                             "Test of G  "lobal Cha  "nges
2
3  /F$1       /F$23      /F$456     /F$78.9    /F$.10123
```

Scripsit Power

The power of Scripsit now comes into play. Let's say you want to change the cell format instructions from a dollar format to a left-justified format. Position the cursor at the beginning of the file and enter the command mode again and type R>/F$>/FL and press ENTER. Press the @ key and R key simultaneously and answer the prompt with 10. Press BREAK and R and ENTER. The command line will say, "REPLACED 4". Simple, wasn't it?

It's true that we could probably have changed our five cell positions from within VisiCalc as quickly, but just think if you had twenty-five or fifty to do! This is just one way in which Scripsit's global change command can be extremely valuable to a VisiCalc user. How about finding all occurrences of a certain command? Rearranging the order of values or labels?

Perhaps you've used Scripsit (or another word processor) and VisiCalc in a helpful combination that you would like to share with other users. Write to me in care of *Basic Computing*. If you desire a response, please enclose a stamped self-addressed envelope.

VisiCalc is a registered trademark of VisiCorp.

TRSDOS 6.0 is copyrighted by Tandy Corp. and Logical Systems, Inc.

Microsoft BASIC is copyrighted by Microsoft Corp.

Scripsit is a registered trademark of Tandy Corp.

Figure 2

```
>E3:/F$.10123
>D3:/F$78.9
>C3:/F$456
>B3:/F$23
>A3:/F$1
>E1:"nges
>D1:"lobal Cha
>C1:"Test of G
/W1
/GOC
/GRA
/GFI
/GC9
/X>A1:>A1:
```

Tandy topics

MC-10 announcement and Model 100 mini program

Ed Juge, Director, Computer Merchandising
1500 One Tandy Center, Ft. Worth, TX 76102

It's hard to believe that it's summer already! Today is Sunday, and I'm sitting on the patio drinking my morning coffee with my Model 100 in my lap. At Tandy, this is our busiest time of year. We've been working on our fall catalogs since March and we're getting ready for our annual regional meetings. Every weekend during August, our buyers will be on the road. We'll be telling our store managers about the fiscal 1984 product lines and they'll be telling us what they need in products, services, and so forth. It's always interesting and hectic.

New MC-10

If you read the *Wall Street Journal* on May 24, you saw our new "Micro Color Computer," Model MC-10. It's the last in our series of six new product announcements, and our entry into the "low-end" home computer marketplace, at only $119.95. It has a real keyboard, with keys that move, color, and 4K of memory, expandable to 16K. Included are cassette and serial ports. Of course, it connects to your own TV and it's small. In keeping with our new look, it's Tandy off-white. (Technically it's fawn gray. Now you can't say I never told you.) So, if you would like a very low-cost TRS-80 starter system to recommend to a friend, or pick as a great gift for a youngster, please be our guest!

We had two more firsts this spring. The first meetings for a newly-formed independent Tandy users' group for large-business users, and an independent TRS-80 legal users' group were scheduled for May at the Americana-Tandy Center Hotel right here in our backyard. A number of Tandy staffers spoke and participated in panel feedback sessions.

Since Cam Brown indicated his (and your) interest in how your favorite computers are being received by the world, maybe you would enjoy this story. A couple of months ago, the Texas Bar Association offered a conference session on computers. A lawyer friend of ours, Rick Rodgers, who now teaches law at Campbell University, organized it, with about twenty TRS-80's. Notices were mailed stating that attendance would be limited to 60. Four days later they were overbooked! To condense a long story, a second overflow session was also overbooked and at both seminars there was a standing crowd. People came from miles away, even though they had been turned down. It was an overwhelming success and will probably be repeated in other states.

How Are Those New Products Doing?

Reaction to our new products has been great! The color change has been welcomed as long overdue. The coming availability of CP/M Plus has pleased everyone, even the trade press. The additional expansion capabilities of the Model 12, green screen, card cage and new keyboard have made it a real winner. Of course, you know by now that the 16 has been restyled as a 16B. It now looks like, and expands like, the Model 12. Xenix is getting absolutely rave reviews from users for its power and speed. The 12 and 16 now represent the only computer family I know of that allows you (at least with our software) to start with a single-user 8-bit machine, upgrade to 16-bit single-user, then to multi-user, maintaining software compatibility all of the way.

I've received several phone calls from new Model 4 owners, just to say "thanks for bringing out such a great computer." The ability to run Model III software as-is, and to upgrade your Model III to a 4, are unique. The 80x24 screen, sound, 4MHz clock and new keyboard are very popular, but it's the TRSDOS 6.0 operating system that really shows off the power of the new computer. If you haven't seen it run, do it! Major accounts are really excited about Model 4. It's an ideal, very powerful workstation, and they get five Model 4s for the cost of one Lisa.

Even our new PC's are looking exceptional. I've even heard from one college that now requires a PC4 before they'll register a student for engineering classes.

What can I say about Model 100? Large companies are buying them in quantity for their traveling staff members. *InfoWorld* reported taking one to an IBM press conference and "stealing the show

from IBM." A vendor reports that the chief executive officer of "a Tandy competitor" carries one in his briefcase and raves about it. Three publishers we know of (and two more rumored) are starting magazines devoted exclusively to the Model 100! Both *PC* and *PC World* (IBM PC magazines) called, wanting to do stories on it. Many computer owners consider the 100 their most useful peripheral to their existing system. There's even a new CompuServe SIG (special interest group) for Model 100 owners, and special Model 100 sections on several others. Yep, Model 100's doing well, too.

So, Is That It?

Seven new models (PC-3, PC-4, 100, 12, 16B, 4, and MC-10) out of our current line of nine TRS-80's isn't too bad. Are we through? Of course not! For example, if you haven't seen the new DMP-120, it's a DMP-200 without the word processing mode. The result is a great personal printer for just $499.95. It makes a 32K Model 100 system with printer less than $1,700, and a 2-disk, 64K Model 4 system under $2,600.

Compatibility

All of the new models have caused several people to ask why we don't make them each software compatible. We try to design specific models to be top performers for specific classes of users. Both software and hardware are optimized and in the process some incompatibilities are inevitable. If you buy the right computer for your job, it usually doesn't matter. Super Scripsit in your Model 100 would leave no memory for anything. Nor would you consider Model 100's Text program an adequate full-time word processor on your Model 4 or 12. And, although I guess you could travel with it, try stuffing your Osb . . . . uh, I mean Model 4 in your briefcase, or using it in a taxi.

I've held my writing down a little this month, so I could pass along a short program for Model 100 owners. It allows some minimal output-formatting of freeform text files. It should give you a little more word processing capability. In fact, I used it for this month's column.

Well, enough of this for one Sunday morning! See you next month.

Model 100 Program Listing

```
1 'NEWPRT.100:ej:04/24/83
2 '<GRPH>p in text forces
end of page
3 'Defaults for margins (L
,R), print
4 '   title on page 1 (HD$
) and line
5 '   spacing (LS) are in
line # 20.
6 '
7 '
10 CLS:CLEAR 2000:FILES
20 L=20:R=60:HD$="Y":LS=1
30 C$=STRING$(60,32):LN=0:
PG=1:EN=1
40 GOSUB100:INPUT".DO file
 to print ";N$
50 GOSUB100:INPUT"Enter pa
ge title";H$
60 GOSUB100:INPUT"Title on
 Pg. 1  (Y/N) ";HD$
70 GOSUB100:INPUT"Margins
(L,R)    ";L,R:IFL=0THENL=1
80 L$=STRING$(L-1,32):GOSU
B100:INPUT"Line Spacing (1
/2) ";LS
90 OPENN$FORINPUTAS1:GOTO1
10
100 PRINT@205,C$:PRINT@205
,"";:RETURN
110 CLS:IFH$=""THENPH$=L$+
DATE$:GOTO140
120 PH$=STRING$(R-8,32)+DA
TE$
130 MID$(PH$,L,(LEN(H$)))=
H$
140 IFHD$="Y"ORHD$="y"THEN
PR$=L$:GOSUB360
160 PR$=L$
170 IFLN>=50THENGOTO330
180 FORJ=LEN(PR$)TOR
190 PR$=PR$+INPUT$(1,1)
200 IFEOF(1)THENCLOSE:EN=2
:GOTO280
210 IFRIGHT$(PR$,1)=CHR$(1
0)THEN270
220 IFRIGHT$(PR$,1)=CHR$(1
28)THENPR$=LEFT$(PR$,J-1):
GOTO330
230 NEXTJ
240 IFMID$(PR$,J,1)=" "THE
N260
250 J=J-1:GOTO240
260 NX$=MID$(PR$,J+1,R):PR
$=MID$(PR$,1,J):GOTO280
270 PR$=LEFT$(PR$,LEN(PR$)
-2)
280 IFMID$(PR$,L,3)="   "T
HEN300
290 IFMID$(PR$,L,1)=" "THE
NPR$=MID$(PR$,2,LEN(PR$)):
GOTO290
300 LPRINTPR$:LN=LN+1:IFEN
=2THEN330
310 IFLS=2THENLPRINT:LN=LN
+1
320 PR$=L$+NX$:NX$="":GOTO
 170
330 FOR J1=LNTO(65*EN):LPR
INT:NEXTJ1:PG=PG+1
340 IFEN=2THENMENU
350 IFX$<>""THEN358 ELSEPR
INT@160,"  <ENTER> = next
page, <C> = continuous":X$
=INKEY$:GOTO350
360 CLS:LPRINTPH$:LPRINTL$
;"Page ";PG
370 LPRINT:LPRINT:LN=4:GOT
0180
```

The secret to LOAD and auto RUN

How to have AUTO in Level II

Models I/III without disks

John Junod

Greenville, IL

I own a 16K Level II without a disk and one thing I often wished I could do was load and run a BASIC program without typing RUN. In the Sept./Oct. 1980, *80-U.S. Journal*, Bill Wilson showed us how to do that if we had a disk, in an article named Teersaty. One of these days that

Listing 1

```
41E2            00100          ORG     41E2H   ;AUTO START
41E2 C36241     00110          JP      START   ;   "
41AC            00120          ORG     41ACH   ;RUN IF BREAK KEY
41AC C36D41     00130          JP      RUN
41B8            00140          ORG     41B8H   ;RUN AFTER CLOAD-
41B8 C36D41     00150          JP      RUN     ;AND RESET BUTTON
4162            00160          ORG     4162H   ;FITS IN DOS EXITS
4162 21E241     00170 START    LD      HL,41E2H; RESET AUTO START
4165 36C9       00180          LD      (HL),0C9H; TO RET
4167 23         00190          INC     HL      ;MAKE THE POSITION
4168 3600       00200          LD      (HL),0  ;MARKED BY HL BE 0
416A C31F2C     00210          JP      2C1FH   ;TO CLOAD IN ROM
416D CDF801     00220 RUN      CALL    01F8H   ;CASSETTE OFF
4170 CD7A1E     00230          CALL    1E7AH   ;CLEAR
4173 2AA440     00240          LD      HL,(40A4H); PGM START POINTER
4176 2B         00250          DEC     HL      ;POINT AT
4177 E5         00260          PUSH    HL      ;THE LEADING
4178 D1         00270          POP     DE      ;ZERO
4179 C31E1D     00280          JP      1D1EH   ;RUN
4162            00290          END     START
00000 TOTAL ERRORS
```

might come to good use for me, but in the meantime I would have to settle for something else.

I finally found a way and wish to share it with all of you in computerland. So, fire up your computer and get your Editor/Assembler up and running and type in the listing. The program, when loaded, will reside in the area where DOS exits EOF, LOC, LOF, MKI$, MKS$ and MKD$ normally are. The program also uses the following DOS exits; 41E2H, 41ACH (called before printing the ready message), 41B8H (called after READY in the CLOAD routine).

The program is actually a loader. Save the loader immediately before a BASIC program using the name of the BASIC program. Now when you wish to use the BASIC program simply type SYSTEM <ENTER>, program name <ENTER>. The loader will load and auto-execute causing the BASIC program to RUN as soon as it finishes loading. The loader also protects your listing since the BASIC program starts over everytime the reset button is pushed or the break key is pressed.

Important: Make sure your BASIC program is debugged and you have another copy. If there is an error in the program it will not let you correct it, unless it is a SN ERROR.

Here is another way to use the same idea. We can now couple your system programs to load as one program. How many times have you tried to type in a very long program into the Editor/Assembler only to find a BUFFER FULL or SYMBOL TABLE OVERFLOW error message pop up? After you have stripped as much as possible and still have an error due to the size of the program, you realize you must split the program into two or more parts. That is not so bad but now it is a little more difficult to run that particular program. The code in Listing 2 will make the first program load the next program on the tape. Start the second to ORG at start's address. You can connect as many system programs as your computer has room for. For example, Listing 3 is a short program I have before Radio Shack's Renum to make the word LINE call up Renum.

Listing 2

```
41E2            00100           ORG     41E2H   ;AUTO START
41E2 C30000     00110           JP      START
0000            00120           ORG     ;(AS NEEDED)
                00130 ;YOUR PROGRAM GOES HERE
0000 21E241     00140 START     LD      HL,41E2H        ;RESET AUTOSTART
0003 36C9       00150           LD      (HL),0C9H
0005 210B00     00160           LD      HL,ALST         ;POINT TO NEXT NAME
0008 C3CE02     00170           JP      02CEH           ;LOAD NEXT PGM
000B 00         00180 ALST      DEFB    0
000C 36         00190           DEFM    '6 LETTERS OR LESS NEXT PGM NAME'
002B 00         00200           DEFB    0
0000            00210           END     START
00000 TOTAL ERRORS
```

Listing 3

```
41E2            00100           ORG     41E2H   ;AUTO START
41E2 C3E641     00110           JP      START
41A3            00120           ORG     41A3H   ;PATCH TO LINE
41A3 C34C7C     00130           JP      7C4CH   ;COMMAND IN DOS
41E6            00140           ORG     41E6H   ;IN INPUT BUFFER
41E6 21E241     00150 START     LD      HL,41E2H;RESET AUTO
41E9 36C9       00160           LD      (HL),0C9H; START
41EB 21F141     00170           LD      HL,ALST ;POINT TO PGM
                00180                           ;NAME
41EE C3CE02     00190           JP      02CEH   ;LOAD NEXT
41F1 00         00200 ALST      DEFB    0
41F2 52         00210           DEFM    'RENUM'
41F7 00         00220           DEFB    0
41E6            00230           END     START
00000 TOTAL ERRORS
```

Basically BASIC

Menu-generating programs

For all models

As we gain experience in programming, we discover that writing modularized programs is easier than just "hacking." And, as our programs become more complex, we find that menu-driven programs are not only easier to write, they are also more friendly to the user.

With this in mind, I developed some routines which automatically generate menus. All the programmer has to do is assign values to a few variables. Here they are, with the hope that you'll be able to adapt the principles involved to some of your programs.

The four listings present variations on a single theme. The first generates a numeric menu, the second an alphanumeric menu. The third and fourth are the first two listings with user-defined functions for input checking.

The Set-up Module

The set-up module (lines 10 to 499) is self-explanatory for most people with even a little programming experience. Line 10 clears 500 bytes of string space. Line 12 defines all variables beginning with X as integers (to speed up FOR... NEXT loops, which use variable XS as a counter).

Lines 152 and 154 assign variables M1$ and M3$ messages to be printed in various parts of the program. Line 172 assigns a line of 63 equal signs (=) to variable L1$. This is one character less than the Model I or III screen width. For Model II, substitute 79 for 63. Color Computer users substitute 31 for 63. Color Computer users without disks should type a string of 31 equal signs for L1$.

Line 499 jumps to the beginning of the main menu at line 1010. I've put the subroutines in the early lines because they operate faster when located there.

The Main Menu

This module begins at line 1000 and assigns values to the variables used by the subroutines which print the menu. The variables used and their purposes are: A$ is the title of the menu (MAIN in this example). A2 is the number of menu items (3 in this example). A$() is an undimensioned array which holds the labels of the menu items. (Because the array isn't dimensioned, there is a maximum of 10 items. If you have more, you should divide them into sub-menus.)

Printing the Menu

The menu printing subroutine begins at line 610 by clearing the screen. Line 612 prints the menu title (MAIN) from variable A$, followed by MENU. Line 614 prints a double line (L1$) across the screen.

Lines 616 to 620 are a FOR... NEXT loop which prints the menu item numbers and names. The final value of the FOR statement (A2, which has been set in line 1010) controls the number of menu item lines to be printed.

Line 618 prints the "<" character as a bracket. CHR$(48+XS) is the number of the menu item. The routine uses the character number 48 + XS (rather than simply printing the value of the counter XS) to eliminate the leading blank which would show if the numeric value of XS were printed. The closing bracket (>) and a space are printed. Following this, the item label is printed from the A$ array.

Line 622 prints another double line (L1$). Line 624 prints the input prompt message from variable M1$ and the INPUT statement assigns the user input to variable R1. Variable A1 is set to 1, the lowest acceptable input value, and the input checking subroutine is called.

Checking the Input

The numeric input check subroutine in line 535, a one-line routine, checks whether the input in variable R1 is within the low (A1) and high (A2) limits. It assigns a true (-1) flag to variable A if the input is bad, or a false (0) flag if it's acceptable. Although this could be done with less apparent effort in the menu printing subroutine, it is a separate subroutine so it can be used by any part of the program which has numeric input to check. This is also the reason that variable A1 is used for the low value. (If you don't understand the logic of A=R1<A1 OR R1>A2 refer to the April, 1983, *Basically BASIC*.)

Program control returns to line 635 in the menu printing subroutine. If A is true (-1, meaning the input is bad) the screen is cleared, the BAD INPUT — REDO message from variable M3$ is printed at the top of the screen, and the entire menu is reprinted. If the input flag in variable A is 0 (meaning the input is good) control returns to line 1020 and the program continues.

An Alphanumeric Routine

Listing 2 uses these same principles for printing menus with alphanumeric indicators rather than numbers for the menu items.

Line 1010 adds B$, which holds the first initial of each menu item label. Each character used can be used only once. The menu printing subroutine begins at line 710 and closely parallels the numeric printing subroutine.

Modular Software Associates introduces . . .

NEWBASIC 2.0

If you program in BASIC, You can't afford to be without NEWBASIC.

Disk BASIC enhancement for your TRS-80 Model I/III

Customize Newbasic

Let's face it. You won't always need every command available with NEWBASIC. That's why we provide the CREATOR program. CREATOR allows you to quickly and easily create a NEWBASIC/CMD module containing only those commands you wish to use, thereby saving memory. There is no limit to the number of NEWBASIC modules you can create, with any combination of commands you desire. CREATOR offers total flexibility combined with ease of use.

Spooler/Despooler

We've seen prices of spooler/despoolers as high as **$100!** We don't agree with this type of pricing philosophy. To prove it, NEWBASIC includes 3 commands to allow true, disk-based spooling and despooling from BASIC. The SPOOLON and SPOOLOFF commands are used to start and stop the spooling of printer output to a disk file. The DESPOOL command provides background **(interrupt-driven)** despooling of a previously spooled print file.

Graphics Commands Galore

Vector graphics, scaling, rotation, circles, ellipses, arcs, lines, boxes, solids . . . We don't know of any package available for the standard TRS-80 Model I/III that offers the wide **variety** and fantastic **power** of our graphics commands. With commands like **CIRCLE, DRAW, LINE, PAINT,** and **PLOT** you can easily draw any shape or figure possible on the TRS-80. Fast. And if it's not fast enough for you, TWO types of **GET/PUT** commands are included for incredible BASIC **animation.**

- **Customize NEWBASIC—include only those commands you need.**
- **Disk-based Spooler/Despooler.**
- **Tremendously powerful graphics commands.**
- **38 pre-defined keys *and* 10 user-defineable keys.**
- **Sound generation and RS-232 commands.**
- **Programming tools for editing and debugging.**
- **Line-labeling, memory move, string execution, DATA restore, and much, much more.**
- **70 page user's manual with scores of examples.**
- **Compatible with most major DOS's for TRS-80 Models I & III.**

Pre-Defined & User-Defineable Keys

With QUICKEY, two keystrokes can give you any one of 38 pre-defined BASIC keywords. DEFKEY allows you to define any 10 keys to be any string you'd like, up to a total of 220 characters for the 10 keys. You can also save the defined keys to disk, or load them in from disk, even under program control.

Sound Generation & RS-232 Control

Now your TRS-80 can produce **sound effects** from BASIC. All you need is a small speaker attached to the cassette cable, the SOUND command, and imagination. For more serious applications, commands for RS-232 initialization and input/output are provided. You can even write a **terminal program** in NEWBASIC!

Programming Tools

The LOC. command locates succeeding instances of a given string or BASIC token. It is an invaluable aid when editing a large program. A new trace command is provided which lists the line being executed, as well as the current value of any variable(s) you select. A calculator-like command is available which displays the answer in 5 different representations, including hex, ASCII, and lsb,msb.

Other Features

Restore the DATA pointer to any DATA statement. Execute a string as a BASIC program statement. Call assembly routines without using DEFUSR. Reference lines by label or the value of a variable. Block move memory to scroll the screen right or left, or copy an array to another array —FAST! All this and **much** more is possible and easy to do with NEWBASIC.

Documentation

We don't skimp. You'll receive a thorough, quality user's manual with plenty of examples. You'll also get a handy reference card with a command summary. And several sample programs which demonstrate the power and simplicity of NEWBASIC.

Compatibility

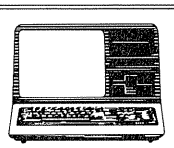
NEWBASIC runs on a 48K RAM, 2+ disk-drive, TRS-80 Model I/III computer. NEWBASIC is compatible with most major DOS's, including DOSPLUS, LDOS, NEWDOS, and TRSDOS. It loads in low memory, below BASIC programs, so it does not disturb any high memory routines you may use.

Who Are We?

MSA was formed in 1980, but you probably haven't heard of us unless you own an LNW80 computer (we wrote LNW's enhanced disk BASIC). Our goal is to offer computer users outstanding software value for their money. We feel NEWBASIC is an excellent realization of this goal.

Pricing and Ordering:

NEWBASIC version 2.0 is only $39.95 (CA residents add 6%) plus $2.50 shipping. Add $2.00 for C.O.D. orders. Foreign orders please add $5.00 shipping (US funds only). NEWBASIC comes with both Model I and III versions on a 35-track, single density, Model I format, data diskette. (2 drives, 48K required.) To order, send check, money order, or credit card (Visa, M/C) information to:



Modular Software Associates

209 18th Street
Huntington Beach, CA 92648

VISA MasterCard

Please allow 2 weeks delay for personal checks. Credit card or C.O.D. orders may be phoned in by calling (714) 960-6668 (weekdays, 8 a.m. - 5 p.m. PST).

Trademarks: TRS-80, TRSDOS (Tandy Corp); NEWDOS (Apparat Inc.); DOSPLUS (Micro-Systems Software), LDOS (Logical Systems Inc.), LNW80 (LNW Research Corp)

Call or Write Your Nearest Snappware Distributor

MICRO-80
284 Goodwood Road
Clarence Park
Adelaide South Australia
Ph-(08) 2117224

DIGI-TEK SYSTEMS
65 Thornridge Circle
Kitchener Ontario N2M-4V9
Ph-(519) 742-8205

STRAWFLOWER ELECTRONICS
50 North Cabrillo
Half Moon Bay
California 94019
Ph-(415) 726-9128

COMPUTER MAGIC
115 Wiltshire Avenue
Louisville, Kentucky 40207
Ph-(502) 893-9334

E-C DATA
Tornevangsvej 88
P.O.B. 116
DK-3460
Birkerod, Denmark
Ph +45/2/81/ 81 91

SYSTEM SOFT
49, Dunvegan Drive
Rise Park
Nottingham, England
NG5 5Dx
Ph-(0602) 275559

CUSTOM COMPUTING
104 Bushwick Rd.
Poughkeepsie, NY 12603
Ph-(914)-471-9318

AEROCOMP
Redbird Airport
Hanger 8
Dallas, Texas 75232
Ph-(214) 339-5104

NOTICE

Snappware Goes On Trial! You Be The Judge!

Snappware knows your programming time is valuable. That's why we are offering a trial package that will cut your programming time up to 75%!

This unprecedented offer allows you to judge for yourself the value of our software using *your* hardware at *your* convenience. Our five best selling products: EXTENDED BASIC, EXTENDED BUILT IN FUNCTIONS, COLLEGE EDUCATED GARBAGE COLLECTOR, AUTOMAP and AUTOFILE, are all available to you on a trial basis for only $50.00 for the Model II and $35.00 for the Model III. We're convinced that after you see how well our software helps you perform programming tasks, you will purchase them like thousands of others have.

Our trial package consists of a master diskette which may be used to create one working copy. Your purchase price for the trial package will be credited toward the purchase of any software Snappware sells.

If for any reason you are not satisfied, just return the trial diskette and working copy and we will refund your money, no questions asked. With an unconditional guarantee like this, you can't lose.

Call our toll free number 800-543-4628 to put us on trial. We're sure your verdict will be: Snappware saves you time and money.

MODEL II Trial Package $50.00
MODEL III Trial Package $35.00

SNAPPWARE

Time saving power at your fingertips.
CALL TOLL FREE:
1-800-543-4628
OHIO RESIDENTS CALL COLLECT: (513) 891-4496
3719 Mantell
Cinti., Ohio 45236

DEALER INQUIRIES WELCOME

contact

POWERSOFT

11500 Stemmons Expressway
Suite 125
Dallas, Texas
75229

CALL:

214-484-2976
Texas residents

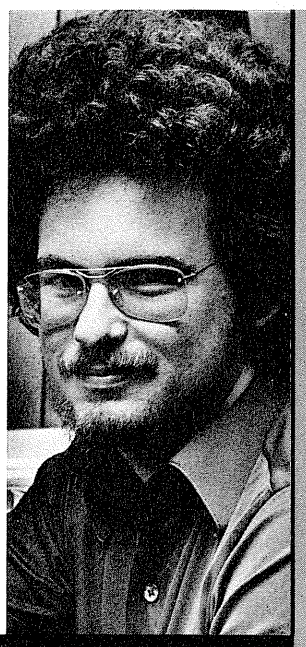
CALL TOLL FREE

800-527-7432

"I BOUGHT IT"

"My biggest loss of programming time using Snappware's COLLEGE EDUCATED GARBAGE COLLECTOR is spent inserting my diskette."

SCOTT ADAMS - PRES. OF ADVENTURE INTL.



The Snappware College Educated Garbage Collector (SNAPP-VI) is an intelligent processing function which greatly improves performance of typical BASIC applications. And here's why.

Microsoft uses a 'variable length string' in the BASIC interpreter. Each time the string is assigned a new value, it is relocated in a string pool. Periodically the string pool must be reorganized and condensed into a single contiguous area. Performing this string space reclamation is time consuming and inefficient because this approach evaluates and collects each string individually. The time required is roughly proportional to the square of the number of active strings in the resident program. During reclamation the system seems to 'lock-up' and does not respond to the operator until the process is completed.

This time consuming approach requires a better solution. Snappware has developed a solution which takes advantage of the auxiliary memory available. SNAPP-VI requires only four bytes per active string as a work area. When free storage space is available, our system temporarily borrows, uses and returns the space to the free storage pool when completed. If storage is not available, our system will temporarily transfer out to disk enough of the BASIC program to make room for our work area and return the 'paged out' information to its correct location when completed.

Benchmarked times show, in some situations, SNAPP-VI performs one hundred times as fast as the Microsoft approach.

If you consider your programming time to be worth money, call us and let us show you how to get more of it.

MODEL II $100.00
MODEL III ..... $ 75.00

*TRSDOS™ Tandy Corporation

SNAPPWARE

Time saving power at your fingertips.

CALL TOLL FREE:

1-800-543-4628

OHIO RESIDENTS CALL COLLECT: (513) 891-4496

3719 Mantell
Cinti., Ohio 45236

The MID$ function in line 718 selects the character from B$ to be printed in the <> brackets. The input character is assigned to variable R$ in line 724.

Checking Alphanumeric Input

Line 542 initializes variable R1 to 0. This 0 will be replaced by the position number in B$ of R$ if the input is acceptable. If the input is bad, R1 will remain with a 0 value and set a bad input flag in line 550.

Line 546 checks the R$ input against each character of B$ to see if it is contained in B$. If it is, the position number of R$ in B$ (XS) is assigned to R1. The counter variable XS is assigned the final value of A2 to speed up the execution of the loop. Line 550, the same numeric test as in the numeric routine, returns a -1 flag if the input is bad, or a 0 flag if it's good.

User-defined Functions

Here's the fun part if you have Disk BASIC. Listing 3 shows the numeric routine amended to use a user-defined function instead of the numeric input checking subroutine. (For a refresher on DEFFN, see the February and March, 1983, *Basically BASIC* columns.)

A user-defined function in line 22 replaces the numeric input checking subroutine in lines 535 and 536 of Listing 1. It returns the same -1 bad flag or 0 good flag that the subroutine did. I've used variables R1, A1, and A2 in the definition argument to show which values are passed to it when the function is called.

Line 624 has been shortened to eliminate the GOSUB (replaced by DEFFN) and the assignment of the low limit of 1 to variable A1 (done in the calling argument). Line 635 calls the function, passing the values from variables R1 and A2, and passing the low value of 1 to the A1 variable in the definition argument.

Note to Color Computer users with Disk BASIC: The Color Computer allows only a single variable to be passed. Change the following lines:

22 DEFFN BN(A1) = R1<A1 OR R1>A2
635 IF FN BN(1) THEN CLS : PRINT M3$: GOTO 612 ELSE RETURN

DEFFN for Alphanumeric Input Check

Listing 4 is the same alphanumeric input check as Listing 2, but with a user-defined function in line 20 replacing the input checking subroutine in lines 542 to 552. It's tricky, so be sure you understand it: DEFFN BI(B$,R$,A1,A2) = INSTR(B$,R$) <A1 OR INSTR(B$, R$) >A2.

The definition argument picks up the values of B$, R$, and A2 passed from the call in line 735. The call also passes the low value of 1 to A1 in the definition argument. The definition expression uses the INSTR function to return the value of the position of R$ in B$ (if R$ isn't in B$, it will return a 0). This value is checked against the high and low limits of A1 and A2 and a bad input flag of -1, or a good input flag of 0, is returned.

Line 735 has been changed as noted. Line 1020 replaces variable R1 with the INSTR function.

Bells and Whistles

These are bare bones examples. After you enter and test them, you will probably want to add your own embellishments. For starters, I'll suggest adding some of Spencer Hall's *Z-Subs* (*80-U.S. Journal*, January, 1982) for such things as centering the title. Perhaps you'll want to speed up the input by adding an INKEY$ routine. After all, I don't know many programmers who can't create several "bells and whistles" to spruce up a program. That's BASIC.

Listing 1

```
Ø 'MENUGEN1/BAS  (A SAVE)  V.2.Ø  Ø5/Ø6/
83  D.55
1 'LISTING 1
2 'Numeric Menu Generator
9 '===== Set-Up Module =====
1Ø CLEAR 5ØØ
12 DEFINT X
15Ø '=== Messages ===
152 M1$="<ENTER> YOUR CHOICE"
154 M3$="BAD INPUT - REDO"
17Ø '=== Lines ===
172 L1$=STRING$(63,"=")
499 GOTO1Ø1Ø
5ØØ '===== Subroutines =====
53Ø '=== Numeric Input Check Subroutine
===
532 'Uses:  R1=Response, A1=Low, A2=High
    From Menu Print Sub
533 'Returns A=Bad Input Flag (-1) or Go
od Input Flag (Ø)
535 A=R1<A1 OR R1>A2
536 RETURN
6ØØ '=== Numeric Menu Print Sub ===
6Ø2 'Uses: A$=Menu Name, A2=# Menu Items
, A$()=Menu Items
          From calling routine
6Ø3 'Inputs R1 User Input
61Ø CLS
612 PRINT A$;" MENU"
614 PRINT L1$
616 FOR XS=1 TO A2
618   PRINT "<"; CHR$(48+XS); "> "; A$(X
S)
62Ø NEXT XS
622 PRINT L1$
624 PRINT M1$; : INPUT R1 : A1=1 : GOSUB
 535
63Ø '== Input Check ==
632 'Uses Input Flag (-1=Bad, Ø=Good) fr
om Input Check Sub
635 IF A THEN CLS : PRINT M3$ : GOTO 612
 ELSE RETURN
1ØØØ '===== Beginning Menu =====
1ØØ2 'Set:  A$ = Menu Title
            A2 = Number of Menu Items
            A$()= Menu Item Labels
1Ø1Ø A$="MAIN":A2=3
```

```
1012 A$(1)="FIRST CHOICE":A$(2)="SECOND
CHOICE":A$(3)="THIRD CHOICE"
1016 GOSUB 6110
1020 ON R1 GOTO 2000, 3000, 4000
2000 PRINT"FIRST CHOICE":STOP
3000 PRINT"SECOND CHOICE":STOP
4000 PRINT"THIRD CHOICE":STOP
```

Listing 2

```
0 'MENUGEN2/BAS  (A SAVE)  V.2.0  05/06/
83  D.55
1 'LISTING 2
2 'Alphanumeric Menu Generator
9 '===== Set-Up Module =====
10 CLEAR 500
12 DEFINT X
150 '=== Messages ===
152 M1$="<ENTER> YOUR CHOICE"
154 M3$="BAD INPUT - REDO"
170 '=== Lines ===
172 L1$=STRING$(63,"=")
499 GOTO1010
500 '===== Subroutines =====
540 '=== Alpha Input Check Subroutine ==
=
541 'R$=Response, A1=Low, A2=High, A=Bad
 Input Flag (-1)
542 R1=0
544 FOR XS=1 TO A2
546  IF R$=MID$(B$,XS,1) THEN R1=XS : XS
=A2
548 NEXT XS
550 A=R1<A1 OR R1>A2
552 RETURN
700 '=== Alpha Menu Print Subroutine ===
702 'Uses:  A$=Menu Title, A2=# Menu Ite
ms, A$()= Menu Item Labels
          B$=Initials of menu choices
710 CLS
712 PRINT A$;" MENU"
714 PRINT L1$
716 FOR XS=1 TO A2
718   PRINT"<"; MID$(B$,XS,1); "> "; A$(
XS)
720 NEXT XS
722 PRINT L1$
724 PRINT M1$; : INPUT R$ : A1=1 : GOSUB
 542
730 '== Input Check ==
735 IF A THEN CLS : PRINT M3$ : GOTO 712
 ELSE RETURN
```

```
1000 '===== Beginning Menu =====
1002 'Set:  A$ = Menu Title
            A2 = Number of Menu Items
            B$ = Initials of Menu Items
            A$()= Menu Item Labels
1010 A$="MAIN":A2=3 : B$="FST"
1012 A$(1)="FIRST CHOICE":A$(2)="SECOND
CHOICE":A$(3)="THIRD CHOICE"
1016 GOSUB 710
1020 ON R1 GOTO 2000, 3000, 4000
2000 PRINT"FIRST CHOICE":STOP
3000 PRINT"SECOND CHOICE":STOP
4000 PRINT"THIRD CHOICE":STOP
```

Listing 3

```
0 'MENUGEN3/BAS  (A SAVE)  V.2.0  05/06/
83  D.55
1 'LISTING 3
2 'Numeric Menu Generator - With DEFFN
9 '===== Set-Up Module =====
10 CLEAR 500
12 DEFINT X
22 DEFFN BN(R1,A1,A2) = R1<A1 OR R1>A2
 'Bad Number
150 '=== Messages ===
152 M1$="<ENTER> YOUR CHOICE"
154 M3$="BAD INPUT - REDO"
170 '=== Lines ===
172 L1$=STRING$(63,"=")
499 GOTO1010
500 '===== Subroutines =====
600 '=== Numeric Menu Print Sub ===
602 'Uses: A$=Menu Name, A2=# Menu Items
, A$()=Menu Items
          From calling routine
603 'Inputs R1 User Input
610 CLS
612 PRINT A$;" MENU"
614 PRINT L1$
616 FOR XS=1 TO A2
618   PRINT "<"; CHR$(48+XS); "> "; A$(X
S)
620 NEXT XS
622 PRINT L1$
624 PRINT M1$; : INPUT R1
630 '== Input Check ==
632 'FN BN Returns -1=Bad Input, 0=Good
Input
635 IF FN BN(R1,1,A2) THEN CLS : PRINT M
3$ : GOTO 612 ELSE RETURN
1000 '===== Beginning Menu =====
1002 'Set:  A$ = Menu Title
            A2 = Number of Menu Items
            A$()= Menu Item Labels
1010 A$="MAIN":A2=3
1012 A$(1)="FIRST CHOICE":A$(2)="SECOND
CHOICE":A$(3)="THIRD CHOICE"
1016 GOSUB 610
1020 ON R1 GOTO 2000, 3000, 4000
2000 PRINT"FIRST CHOICE":STOP
3000 PRINT"SECOND CHOICE":STOP
4000 PRINT"THIRD CHOICE":STOP
```

Listing 4

```
0 'MENUGEN4/BAS  (A SAVE)  V.2.0  05/06/
83  D.55
1 'LISTING 4
2 'Alphanumeric Menu Generator - With DE
FFN
9 '===== Set-Up Module =====
10 CLEAR 500
12 DEFINT X
20 DEFFN BI(B$,R$,A1,A2) = INSTR(B$,R$)
<A1 OR INSTR(B$,R$) >A2
150 '=== Messages ===
152 M1$="<ENTER> YOUR CHOICE"
154 M3$="BAD INPUT - REDO"
170 '=== Lines ===
172 L1$=STRING$(63,"=")
499 GOTO1010
500 '===== Subroutines =====
700 '=== Alpha Menu Print Subroutine ===
702 'Uses:  A$=Menu Title, A2=# Menu Ite
ms, A$()= Menu Item Labels
          B$=Initials of menu choices
710 CLS
712 PRINT A$;" MENU"
714 PRINT L1$
716 FOR XS=1 TO A2
718   PRINT"<"; MID$(B$,XS,1); "> "; A$(
XS)
720 NEXT XS
722 PRINT L1$
724 PRINT M1$; : INPUT R$ : A1=1
730 '== Input Check ==
735 IF FN BI(B$,R$,1,A2) THEN CLS : PRIN
T M3$ : GOTO 712 ELSE RETURN
1000 '===== Beginning Menu =====
1002 'Set:  A$ = Menu Title
            A2 = Number of Menu Items
            B$ = Initials of Menu Items
            A$()= Menu Item Labels
1010 A$="MAIN":A2=3 : B$="FST"
1012 A$(1)="FIRST CHOICE":A$(2)="SECOND
CHOICE":A$(3)="THIRD CHOICE"
1016 GOSUB 710
1020 ON INSTR(B$,R$) GOTO 2000, 3000, 40
00
2000 PRINT"FIRST CHOICE":STOP
3000 PRINT"SECOND CHOICE":STOP
4000 PRINT"THIRD CHOICE":STOP
```

Ammicro introduces the first letter quality printer for $680 that can also be used as a typewriter.

The MICROWRITER™ Daisy wheel printer.

There was a need for a low cost letter quality machine that would be suitable for use as an office typewriter, and as a computer printer. Ammicro met that need by combining the Microwriter parallel interface and the traditional Olivetti craftsmanship that was available in their Praxis machine.

With the Microwriter you can have the best of both worlds a letter quality printer, and a high quality office typewriter all in one machine, that sells for less than the cost of a good dot matrix printer!

It's not just printer or a typewriter that comes complete with a deluxe carrying case, but a feature-packed, lightweight machine that doubles as an office typewriter. This printer is a simple, low cost, reliable unit which can be utilized with word processing systems, microcomputers, personal computers, and small business systems. The Microwriter's low noise level and slim modern styling allow it to blend with any decor.

The Microwriter's print quality is identical to the finest office typewriters on the market. This machine is not only perfect for letters and manuscripts, but with it s 165 character, 12 inch print width, the machine is perfect for letter quality budget spread sheets, price lists, data sheets, and forms.

The Microwriter can tab, rule single lines both vertical and horizontally, underline and print at 10, 12, or 15 characters per inch (switch selectable)! Its ten character memory for automatic error correction, lift off correction ribbon, and fixed or programmable page formats are a few of the many features that make it a perfect office typewriter. Microwriter not only handles letter and legal size sheet paper in widths up to 12 inches wide, but also handles fanfold paper.

There's a wide selection of 21 interchangeable daisy wheels available. And ribbon cassettes that just drop in. With the Microwriter you will never again have to send an important letter or a simple correspondence that doesn't look impressive.

It s operation as a computer printer is simple. Just load it up with paper and you are ready to go. Centronics compatible parallel output cables are currently available from stock for the following computers: IBM PERSONAL COMPUTER™, OSBORNE 1™, ZENITH Z-100™, BURROUGHS B-20™, Convergent Technologies models IWS & AWS™, TRS-80 MODEL I, II, III™, APPLE II™. . . custom cables also available by special order.

This machine creates a new standard by which all current low cost letter quality printers will follow. Ammicro's Microwriter is truly designed for the lifestyles of the 80's and for decades to come.

Why settle for just any printer when you can have a MICROWRITER. . . . a fine letter quality typewriter for you and your computer.

The Microwriter is the only daisy wheel printer on the market for $680. For more information, see your local computer dealer or contact Ammicro directly.

ammicro™ corp

122 East 42 Street, Suite 1700, New York, N.Y. 10168
(212) 254-3030

DEALER INQUIRIES INVITED

MasterCard OR VISA

For orders call:
1-800-251-5110

MICROWRITER is a trademark of Ammicro Corporation. PARAXIS 30 is a trademark of Olivetti

The love test

Find out which type of lover you are

Color Computer

Michael Nugent, Ph.D., Kent, WA

"I still *love* you, but I'm not *in love* with you. Do you understand?"

"Yes, I understand. *You don't love me anymore!*"

As a marriage and family therapist, I frequently hear couples sincerely trying to communicate about their love and being frustrated by the very language they use. In English, people can "love" their spouses, ice cream, and the poor of Bangladesh! The LOVE Test provides a better way to talk about this most important human experience. When this understanding is shared with a partner, both can accept the love of the other for what it is.

About Love

Researchers have found six main types of love and they use a combination of Greek and Latin words to describe them: Storge (pronounced stor'-gay), Agape (pronounced a-gaw'-pay), Mania, Pragma, Ludus and Eros. The graphic display created by this program represents the relative importance to you of each of the six main types according to your responses on The LOVE TEST (the longest line corresponds to your preferred type, the shortest to your least preferred type).

Storge is the love of long-time friends. It is deep, stable and confident. There is a deep affection and general liking. Typically, Storge lovers do not "fall in love" but, rather, their friendship just keeps getting more and more rich and satisfying. When Storge lovers marry, they expect to be treated with respect and fairness, for that is what they will extend to others. The partner of a Storge lover is considered that person's best friend and confidant.

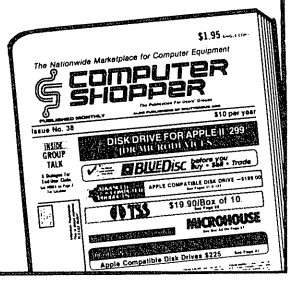
Agape is often referred to as "love from afar" because the Agape lover needs no direct contact with the object of his or her affections. This totally unselfish approach is rare and may cause problems if the one who is loved is also of the Agape variety since each would be trying to give (unselfishly) and there would, therefore, be no one to receive gifts in the relationship. Agape is the word used in the New Testament to describe God's love for man. It is not romantic or sexual unless, of course, those are the desires of the beloved.

Mania is identified by huge mood swings. Euphoric when together and miserable when apart, these lovers feel in love only when in the emotional peaks and valleys. While this form of love is typical of a "first love," it is not confined to adolescents. The heady "I'm in love with love" feeling has a fairly short life since the stress of constant mood swings takes its toll on the relationship. Furthermore, the jealousy of Mania constantly tests the relationship and this further stresses it.

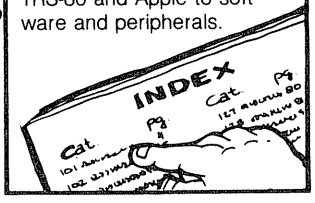
Pragma is "love with a shopping list." A job description is first written (consciously or not) and then only those who meet the criteria can be loved. While not romantic, Pragma lovers can expect long-term relationships as long as their values do not change, thereby making their partners no longer desirable. Often, Pragmatic love grows over the years as the wisdom of the choice is fully realized.

How to Buy or Sell Computer Equipment and Software

START HERE WITH...



Begin your search in the index of Computer Shopper's bargain filled pages. Locate the category and page number of items that interest you from TRS-80 and Apple to software and peripherals.

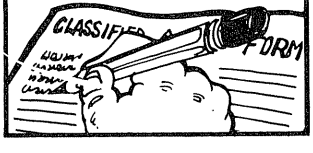


Start or add to your computer system by finding money saving bargains in each month's issue from individuals who no longer need their personal equipment.

You've got your computer hardware, but what about the software? Use a Computer Shopper ad to find what you need. Someone advertising in Computer Shopper probably has what you want.

If you need help with any computer related problem whether it's an interface problem or advice on the right peripheral for a home-brew system, use the free HELP column especially designed for that purpose.

As you outgrow your system or want to trade up (most dealers won't take trade-ins), use Computer Shopper ads to sell your items to 20,000 readers nationwide for the low cost of 12 cents per word.



Computer Shopper is THE nationwide publication for buying, selling and trading Micro and Mini-computer equipment and software. Each issue has over 100 pages full of bargains of new and used equipment.

You can save hundreds of dollars by getting the equipment you need from the hundreds of classified ads individuals place in Computer Shopper every month.

Now is the time for you to join over 20,000 other computer users who save time and money with a subscription to Computer Shopper.

Subscribe today and get your first issue and a classified ad absolutely FREE. Type or print your ad on a plain piece of paper and send it along with your subscription.

Just fill in the coupon or MasterCard or VISA holders can phone *for faster service* and start making your computer dollar go further today.

Cut out and mail to: **COMPUTER SHOPPER**
P.O. Box F534 ● Titusville, FL 32780

Yes, I'll try Computer Shopper, I understand that if I'm not satisfied with my first issue I can receive a full refund and keep the first issue free.

☐ 1 year $10.00 ($30.00 in Canada)

☐ I have enclosed my free classified ad.

☐ I want to use my free ad later, send me a coupon.

NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____ ZIP: _____

COMPUTER SHOPPER
P.O. Box F534 ● Titusville, FL 32780
Telephone: 305-269-3211

Ludus lovers are playful. As a result, they enjoy the chase more than the relationship that might develop if they settled down. They sometimes deliberately "play hard to get" to entice a lover and may even invent an imaginary person to keep the beloved from becoming too confident about the relationship. Ludus lovers reveal deep feelings only reluctantly because they are extremely threatened by being vulnerable in an emotionally intimate way. These lovers care a great deal about their sexual performance and the performance of their partners, and are very conscious of what others might think of their choice of partners.

Eros is immediate, visual, and romantic. "Falling in love at first sight" is typical because these lovers have an ideal image of beauty in their minds at all times; When someone matches that picture, they know, "This is the real thing!" All the senses are involved, as this type emphasizes the romantic value of aromas, music and other sounds, the taste of the kisses, and the touch of the beloved. Sex is "experienced" not "performed." With such a focus on the immediate beauty of the beloved, Eros lovers can fall slowly out of love as their partners grow old, have medical problems, or simply change their "look."

Combinations of several styles are more common than pure types since people in general are complex rather than simple. Certain combinations seem especially valuable in a long-term relationship. For example, an Eros-Storge combination would allow both the immediate rush of the feeling associated with "true romance" and "chemistry" and the long-term growth potential of a Storge friendship. Add a bit of Pragma to screen out potential partners whose characteristics make them poor prospects for a successful marriage, enough Agape to provide care and nurturing when the partner is ill, and you can make plans for a 50th anniversary party.

Other considerations are presented at the bottom of the response display. These include the preferred pace of a relationship, the center (or focus) of the relationship, and the mode of experiencing love. For each, the absolute amounts are less important than the degree of balance between the two extremes. Being too far to the left or right of center on these scales can be detrimental to the formation and maintenance of love relationships.

For example, people who strongly prefer a fast pace can frighten those with a slower pace. Too much self-interest can easily destroy a relationship, but too little can also be dangerous by making the partner feel unnecessary and unvalued. Similarly, a balance between emotion and rationality is the best predictor of success in any intimate relationship.

Each of the styles has advantages and limitations. Relationships seem to be most stable when both partners share a common understanding of what love is, or at least can respect and accept the definition used by their partners.

About This Program

This program is written in Extended BASIC for the Radio Shack Color Computer. It incorporates a graphics title page (lines 190 to 490), a compact item-administration routine (lines 560 to 640), and a graph-generating display section (lines 690 to 1050). Lines 1450 to 1630 form an option to create a permanent copy of the relative scores on a standard 80-column printer (a Radio Shack Line Printer VIII was used for development, but no codes unique to that printer were used in the program).

The LOVE Test is the first computer program of a series called "Understanding People." To be notified of each program as it becomes available, write to the author at Counseling and Preventive Services (CAPS), Oakhurst—Suite 121, 1851 S. Central Place, Kent, WA 98031. CAPS provides marriage, family, and child counseling; preventive seminars; management consulting; and computer usage instruction. Dr. Nugent is a Clinical Member of the American Association for Marriage and Family Therapy and an assistant professor of Human Behavior for Newport University.

Program Listing for The LOVE Test

```
10 '-------THE LOVE TEST-------
20 '      COPYRIGHT 1982 BY
30 '  MICHAEL D. NUGENT, PH.D.
40 '  DIRECTOR, COUNSELING AND
50 ' PREVENTIVE SERVICES (CAPS)
60 '    OAKHURST - SUITE 121
70 ' 1851 SOUTH CENTRAL PLACE
80 'KENT,WA 98031 (206)854-7072
90 '-------------------------
100 ' THIS PROGRAM IS THE FIRST
110 'IN A SERIES CALLED
120 'understanding people.
130 ' WE WILL NOTIFY YOU OF NEW
140 'PROGRAMS IN THIS SERIES IF
150 'YOU WILL SEND US YOUR NAME
160 'AND ADDRESS.       => MDN
170 '-------------------------
180 FORST=1TO6:ST(ST)=0:NEXT:X0=7:X1=159
:X2=133:X3=138
190 PMODE1,1 '-----TITLE PAGE-
200 PCLS1
210 SCREEN1,0
220 FORK=1TO255STEP10:SOUNDK,1:NEXT
230 FORK=0TO180STEP10
240 LINE(0+K,190-K)-(255-K,0+K),PSET
250 NEXT
260 IFX0=6THEN330
270 COLOR X0,5
280 FORK=180TO0STEP-10
290 LINE(255-K,0+K)-(0+K,190-K),PSET
300 NEXT
310 X0=X0-1:COLOR X0,5
320 GOTO230
330 FORW=1TO500:NEXTW:COLOR7,5
340 DRAW"BM20,20;R5;D10;U10;R5;B;R5;D10;
U5;R5;U5;D10;B;R5;R5;L5;U5;R5;L5;U5;R5"
350 COLOR8,5:PLAY"C;G;O4;C"
360 DRAW"BM20,60;D50;R35;U10;L25;U40;L10
```

The Answer is...

NEWSCRIPT!™

THE WORD PROCESSOR FOR BUSINESSMEN AND PROFESSIONALS

With ongoing support directly from us

A FEW OF NEWSCRIPT's 200 STANDARD FEATURES:

- FORM LETTERS WITH MERGING OF NAMES AND ADDRESSES
- GIVES SUPERB APPEARANCE TO YOUR FINAL DOCUMENTS
- COMPREHENSIVE MANUAL WITH TUTORIAL AND EXAMPLES
- CENTERING, TOP/BOTTOM TITLES, INDENTS, PAGINATION
- UNDERLINING, BOLDFACE, DOUBLE-WIDTH, ITALICS†
- SUB/SUPER SCRIPTS, RIGHT-JUSTIFIED PROPORTIONAL†
- CREATES TABLE OF CONTENTS, SORTED INDEX
- **"LEGAL"** LINE NUMBERING
- SCREEN GRAPHICS, SPECIAL PRINTER SYMBOLS†
- SEARCH/REPLACE GLOBALLY OR WITHIN LINES, COLUMNS
- BLOCK MOVE, COPY, DELETE, INSERT. FILE MERGES
- AUTOSAVE, WHOOPS, DIRECTORY, KILL
- SUPPORT FOR **ALL** LISTED PRINTERS IS INCLUDED ** **(NO PATCHES INVOLVED)** **
- SUPPLIED READY-TO-RUN ON "TINY" DOSPLUS
- ALSO RUNS UNDER NEWDOS/80, LDOS, MULTIDOS, TRSDOS

NEWSCRIPT 7.0:	**$124.95**
Mailing Labels Option:	**29.95**
Special: NEWSCRIPT + LABELS:	**139.95**
Daisywheel Proportional Option:	**49.95**
"Pencil"/"Scripsit" File Convertor:	**24.95**
NEWSCRIPT Manual & Reference card only:	**29.95**
Electric Webster + Correction Feature:	**149.50**
Hyphenation Feature for Electric Webster:	**49.95**
Grammatical Feature for Electric Webster:	**39.95**
Graphics Editor and Programmer (GEAP):	**49.95**
Dotwriter High-resolution graphics:	**69.95**
Special: GEAP + Dotwriter:	**99.95**

REQUIRED CONFIGURATION:

48K TRS-80, MAX-80, LNW, or compatible, with one or more disk drives. Specify Model I or Model III.

† some features work only if your printer has the mechanical capability.

** Daisy Wheel Proportional is an extra-cost option.

BUILT-IN SUPPORT FOR MOST POPULAR PRINTERS†, INCLUDING:

Anadex, Brother, Centronics. C.Itoh, Diablo, Epson, Gemini, Microline, NEC, Prowriter, Qume, Radio Shack (LP 1-8, DW2, DMP-410, DWP 200-2100), Smith Corona, Teletype, Typewriter, anything compatible with any of these, and many others, parallel and RS-232.

SPECIAL AVAILABLE OPTION: Right-justified proportional for Diablo, F-10, Qume, Spinwriter, etc. Requires "Daisywheel Proportional" Option plus NEWSCRIPT.

REVIEWERS AND USERS AGREE:

"NEWSCRIPT" is the best word procesor I have seen . . . unsurpassed in printer control . . . no other TRS-80 word processor can match its ability to format text . . . its editor is fast, easy, and powerful." *(80 MICRO, Oct. 1982)*

"Your phone information system and the prompt and courteous staff that you provide to help your clients . . . are worth the cost of the system." *(V.H.H.)*

"Better than cold beer on a hot day!! Thank you!!" *(R.S.)*

"What a program. So easy to learn and easier to use. I waited too long before ordering!" *(P.J.M.)*

". . . takes the TRS-80 to a new level of text handling . . . very user-friendly . . . superb documentation, adaptability to many printers and operating systems . . . a standard against which other TRS-80 word processing programs will be judged." *(SOFTSIDE, Dec. 1982)*

". . . ongoing support second to none, with superb documentation." *(80 U.S. Journal, Feb. 1982)*

TO ORDER, CALL NOW, TOLL-FREE: (800) 824-7888, Operator 577

For orders, information, or names of nearby dealers: (213) 764-3131, or write to us.

Order from your Software dealer or from:

PROSOFT®

Dep't. C, Box 560 No. Hollywood, CA 91603

TERMS: VISA, Mastercard, checks, money orders, COD. No P.O.'s accepted. Most orders shipped within 24 hours. Please add $3.00 for surface UPS in U.S.A., or $6.00 for UPS Blue Label. Add $6.00 in Canada, $15.00 overseas air shipment, 6½% sales tax in California.

```
":PAINT(24,64),8,8
370 CIRCLE(90,86),29:CIRCLE(90,86),18:PA
INT(90,62),8,8
380 LINE(120,60)-(145,110),PSET
390 LINE(145,110)-(155,110),PSET
400 LINE(155,110)-(180,60),PSET
410 LINE(180,60)-(170,60),PSET
420 LINE(170,60)-(150,100),PSET
430 LINE(150,100)-(130,60),PSET
440 LINE(130,60)-(120,60),PSET
450 PAINT(122,62),8,8
460 DRAW"BM190,60;D50;R30;U10;L20;U10;R1
0;U10;L10;U10;R20;U10;L30":PAINT(192,62)
,8,8
470 COLOR7,5:PLAY"O4;D#;E"
480 DRAW"BM175,150;R10;L5;D10;B;R10;R5;L
5;U5;R5;L5;U5;R5;B;R5;R10;L10;D5;R10;D5;
L10;B;R20;U10;L5;R10"
490 FORW=1TO1500:NEXTW
500 CLS  '----INSTRUCTIONS---
510 PRINT:PRINT" BY RESPONDING TO 36 SEN
TENCES  ABOUT LOVE, YOU CAN DETERMINE
 WHAT TYPE OF LOVE YOU HAVE TO  GIVE, A
ND WHAT TYPE YOU NEED."
520 PRINT:PRINT" FOR EACH ITEM, PRESS t
IF THE  ITEM IS TRUE OR GENERALLY TRUE
 FOR YOU, AND f IF IT IS FALSE  OR GENE
RALLY FALSE."
530 PRINT:PRINT" PRESS enter WHEN YOU AR
E READY  TO BEGIN."
540 SCREEN0,1:SOUND1,1:R$=INKEY$
550 R$=INKEY$:IF R$=""THEN 550
560 FORST=1TO6  '---ITEM ADMIN-
570 FORI=1TO6
580 READA$:CLS:PRINT@33,A$
590 PRINT@439,"T=TRUE":PRINT@471,"F=FALS
E":R$=INKEY$
600 R$=INKEY$:IF R$=""THEN600
610 IF R$="T" THEN ST(ST)=ST(ST)+1:GOTO6
30
620 IF R$<>"F" THEN600
630 SOUND200,1:CLS:FORW=1TO300:NEXTW:NEX
TI
640 NEXTST
650 CLS(3):SOUND1,1
660 PRINT@33," INTERPRET YOUR SCORES ON
THE ";:PRINT@65," FOLLOWING DISPLAY ACCO
RDING  ";:PRINT@97," TO THE LOVE TEST AR
TICLE IN  ";:PRINT@129," BASIC COMPUTING
 (JULY 1983).  ";
670 PRINT@193,"  PRESS enter TO CONTINU
E.  ";
680 R$=INKEY$:IF R$=""THEN680
690 CLS  '--RESPONSE DISPLAY---
700 PRINT"  THE LOVE TEST":PRINT
STRING$(32,255);
710 PRINT"STORGE "
720 PRINT"AGAPE  "
730 PRINT"MANIA  "
740 PRINT"PRAGMA "
750 PRINT"LUDUS  "
760 PRINT"EROS   ":PRINT STRING$(32,255)
;
770 PRINT"  PACE"
780 PRINT"SLOW<<<<<<<<<<<<>>>>>>>>>>>>FA
ST";
790 PRINT"  CENTER"
800 PRINT"OTHER<<<<<<<<<<<<>>>>>>>>>>>>SE
LF";
810 PRINT"  MODE"
820 PRINT"FEEL<<<<<<<<<<<<>>>>>>>>>>>>THI
NK";
830 '-----PLOT RESPONSES------
840 FOR ST=1TO6
850 IF ST(ST)=0 THEN 880
860 LO=71+((ST-1)*32):LE=ST(ST)*4
870 PRINT@LO,STRING$(LE,X1):X1=X1+16
880 NEXT ST
890 FOR ST=1TO6
900 IF ST(ST)=6 THEN ST(ST)=5.5
910 NEXT ST
920 L=336:LO=L-ST(1)*2
930 PRINT@LO,STRING$(ST(1)*2,X2);
940 PRINT@L,STRING$(ST(6)*2,X3);
950 L=L+64:LO=L-ST(2)*2
960 PRINT@LO,STRING$(ST(2)*2,X2);
970 PRINT@L,STRING$(ST(5)*2,X3);
980 L=L+64:LO=L-ST(3)*2
990 PRINT@LO,STRING$(ST(3)*2,X2);
1000 PRINT@L,STRING$(ST(4)*2,X3);
1010 FOR K=335TO463STEP64
1020 PRINT@K,CHR$(128);:PRINT@K+1,CHR$(1
28);
1030 NEXT K
1040 FOR K=1TO255 STEP5:SOUND K,1:NEXT K
1050 PRINT@481,"PRESS break TO END, p TO
 PRINT";
1060 '------------------------
1070 R$=INKEY$:IF R$<>"P" THEN 1070 '---
-----EFFECTIVE END-----
1080 '------------------------
1090 DATA I VALUE MARRIAGE AND FAMILY
  MORE THAN MOST PEOPLE.
1100 DATA I EXPERIENCE LOVE AS STABLE
  AND STEADY RATHER THAN AS  HAVING
 HIGH PEAKS AND LOW  VALLEYS OF EMO
TION.
1110 DATA I NEED TO LIKE A PERSON BEFORE
  I CAN LOVE THEM.
1120 DATA I CANNOT LOVE SOMEONE I DO NOT
  RESPECT AND TRUST COMPLETELY.
1130 DATA I AM STILL FRIENDS WITH FORMER
```

```
 LOVERS.
1140 DATA I USUALLY KNOW A PERSON FOR A
  LONG TIME BEFORE I COME TO      LOVE T
HEM.
1150 DATA I FIND IT EASY TO LOVE SOMEONE
  WITHOUT BECOMING PHYSICAL.
1160 DATA I SEE LOVE AND SEXUAL LUST AS
  TOTALLY SEPARATE EXPERIENCES.
1170 DATA I HAVE LOVED SOME PEOPLE
  WITHOUT THEIR EVEN KNOWING      THAT I
 FELT THAT WAY.
1180 DATA I AM MUCH MORE COMFORTABLE
  IN GIVING THAN IN RECEIVING.
1190 DATA I TEND TO LOVE THOSE WHO NEED
  ME.
1200 DATA MY LOVE FOR MY PARTNER IS THE
  SAME AS MY LOVE FOR ALL PEOPLE.
1210 DATA LOVE INTOXICATES ME - I FEEL
  AN INCREDIBLE JOY AND EUPHORIA.
1220 DATA I OFTEN DAYDREAM ABOUT MY
  LOVER AND I HOPE HE OR SHE      DAYDRE
AMS ABOUT ME.
1230 DATA I AM MISERABLE WHEN I AM AWAY
  FROM MY LOVER.
1240 DATA I AM MORE JEALOUS THAN MOST
  PEOPLE.
1250 DATA BITTERNESS USUALLY FOLLOWS A
  PERMANENT BREAKUP.
1260 DATA I SOMETIMES FIND MYSELF LOVING
  SOMEONE WHO DOES THINGS I       DON'T
LIKE.
1270 DATA I FREQUENTLY EVALUATE THE
  PRO'S AND CON'S OF MY RELA-     TIONSH
IPS.
1280 DATA I THINK IT IS VITAL THAT BOTH
  PEOPLE SHARE MOST INTERESTS.
1290 DATA I WOULD LEAVE WITHOUT DELAY OR
  REGRET IF MY PARTNER TOOK       ADVANT
AGE OF ME.
1300 DATA I HAVE A CLEAR SET OF GOALS
  FOR MY LIFE AT VARIOUS STAGES   (LIKE
BEFORE RETIREMENT).
1310 DATA I WOULD AVOID FALLING IN LOVE
  WITH SOMEONE FROM A DIFFERENT   RELIGI
ON OR RACE.
1320 DATA I WOULD NOT FALL IN LOVE WITH
  SOMEONE WHO HAD HABITS I DID    NOT AP
PROVE OF (LIKE SMOKING).
1330 DATA I WOULD END A RELATIONSHIP IF
  MY PARTNER WAS NOT AN EXCITING  SEXUAL
 PARTNER.
1340 DATA I NEED TO STAY EMOTIONALLY
  INDEPENDENT FROM MY PARTNER.
1350 DATA I LIKE TO FLIRT.
1360 DATA I SOMETIMES DELIBERATELY STAY
  AWAY FROM MY LOVER TO MAKE HIM  OR HER
 WANT ME MORE.
1370 DATA I DON'T LIKE LONG DISCUSSIONS
  ABOUT FEELINGS.
1380 DATA I KEEP SOME FEELINGS HIDDEN
  FROM MY LOVER.
1390 DATA I AM VERY ATTRACTED TO PEOPLE
  WITH A PARTICULAR FEATURE       (LIKE
A CERTAIN HAIR COLOR).
1400 DATA I HAVE FALLEN IN LOVE 'AT FIRS
T SIGHT' AT LEAST ONCE.
1410 DATA I FIND YOUNG PEOPLE MUCH MORE
  ATTRACTIVE THAN OLD PEOPLE.
1420 DATA I NEED TO BE TOUCHED AND
  I ENJOY CUDDLING MORE THAN      MOST P
EOPLE.
1430 DATA I NEED TO FEEL SEXUALLY IN
  TUNE WITH MY LOVER.
1440 DATA I USUALLY COMMIT TO A RELATION
- SHIP QUICKLY OR NOT AT ALL.
1450 PRINT@480," PRINTING, IF PRINTER IS
 ONLINE";  '--<OPTIONAL>------PRINTER RO
UTINE----------
1460 FORK=1TO4:PRINT#-2:NEXTK:PRINT#-2,"
NAME";STRING$(50,".");"DATE";STRING$(20,
".");PRINT#-2:PRINT#-2
1470 PRINT#-2," ----------------------
------> THE LOVE TEST <----------------
-------------"
1480 FORK=1TO2:PRINT#-2:NEXTK
1490 PRINT#-2,"STORGE ";STRING$(10*ST(1)
,">")
1500 PRINT#-2,"AGAPE  ";STRING$(10*ST(2)
,">")
1510 PRINT#-2,"MANIA  ";STRING$(10*ST(3)
,">")
1520 PRINT#-2,"PRAGMA ";STRING$(10*ST(4)
,">")
1530 PRINT#-2,"LUDUS  ";STRING$(10*ST(5)
,">")
1540 PRINT#-2,"EROS   ";STRING$(10*ST(6)
,">")
1550 PRINT#-2:PRINT#-2
1560 PRINT#-2,TAB(38)"PACE"
1570 PRINT#-2,"SLOW";TAB(40-ST(1)*4)STRI
NG$(ST(1)*4,"<");TAB(40)STRING$(ST(6)*4,
">");TAB(75)"FAST"
1580 PRINT#-2,TAB(37)"CENTER"
1590 PRINT#-2,"OTHER";TAB(40-ST(2)*4)STR
ING$(ST(2)*4,"<");TAB(40)STRING$(ST(5)*4
,">");TAB(75)"SELF"
1600 PRINT#-2,TAB(38)"MODE"
1610 PRINT#-2,"FEEL";TAB(40-ST(3)*4)STRI
NG$(ST(3)*4,"<");TAB(40)STRING$(ST(4)*4,
">");TAB(74)"THINK"
1620 FORK=1TO20:PRINT#-2:NEXTK
1630 GOTO1050
```

A fast idiot and a slow genius

The computer and you

Joan Horrigan, Fullerton College
321 East Chapman, Fullerton, CA 92634

The problems which occur during the early stages of learning BASIC stem from failing to grasp that the computer is a fast idiot and the programmer is a slow genius. Another way of perceiving this is to consider oneself the slow master, and it the speedy slave. Beginning programmers like myself are quick to grasp this insight (geniuses that we are) and so the electronic beast is put to work. What else are slaves for?

One such prodigy (should I confess?), while launched deep into that mysterious realm of BASIC known as the FOR . . . NEXT loop, found that the swift step 'n fetch slave was willing and able to perform wonders immediately, upon request. I soon learned that the FOR . . . NEXT loop, when done by the speed demon computer, could complete multitudinous tasks — anything from keeping time for a flashing display, to counting and comparing any amount from minus thousands to plus millions. I decided that this slave was for me, so I set about mastering it and BASIC too.

Having quickly gained new skills, I was indeed feeling like a genius (another phenemenon at which I am an expert). For example, I knew that a genius, having lightening-like mental acuity, could spontaneously survey multiple routines via the imagination and was also able to flash a complex composite plan of action through his mind's eye in a microsecond. Similarly, the computer could cycle and process BASIC commands as expeditiously. Hence, those swift executions of the computer slave inspired my ambition and awakened my budding programming skills.

I felt the rush of excitement that creativity bestows. My program was taking shape right on the spot! Sensing my inborn intellect, I proceeded to put the freshly-acquired BASIC commands to work by pounding ferociously on the keyboard until my program was complete. The revealing of my genius was to be disclosed when I typed the word RUN.

Oh horrors! Stunned, and filled with disbelief, I stared in semi-shock until the realization dawned on me: the speedy slave had revolted! How dare such impertinence!

The message, SYNTAX ERROR IN LINE 140, mocked my intelligence as the script (so cold and impersonal) glared stupidly at me from the video screen. The idiot slave refused to budge.

No appeal would work on the idiot until the genius bowed to its command. What an ironic reversal of roles! What humiliation! What was wrong here? There was no finding out until the offending line was examined character by character to uncover the culprit causing such chaos. Either something had been put in, or left out, that did not conform to the laws of BASIC. Where was that little devil, that of-

THE PROGRAMMER'S GUILD PRESENTS

PAC DROIDS™

BY

CHARLES FORSYTHE

THE ULTIMATE IN PAC-ACTION

ONLY $19.95 **UP TO 4 PLAYERS!!**

Unlike any other "PAC" game you've ever seen!!

Hot Machine Language Multi-Color Hi-Res Graphics For All 16K TRS-80 Color Computers

MORE SOUND-MORE ACTION
MORE FEATURES THAN ANY "PAC" GAME IN EXISTENCE!!

Try "PAC-DROIDS"™ for the Outer Limit in pure, explosive arcade action!!

SEND $19.95 CHECK/MONEY ORDER OR VISA/MC#

THE PROGRAMMER'S GUILD
PO BOX 66
PETERBOROUGH, NH 03458
OR CALL (603) 924-6065 FOR C.O.D.

AND GET "FREE" SHIPPING ANYWHERE ON THE PLANET EARTH OR HER COLONIES

Fast idiot

fending character?

Sometimes genius is slow in determining where such a character is, or is not. One's mentality becomes insulted by such menial tasks, fit only for slaves. Escalation of this thought prodded my indignation, and aided in the invention of new expletives. Like the loop, my ire was incrementing until I had worked myself into quite a wrath.

Whom was I mad at? That was a crucial question. Can a sane adult be angry with a mere plastic box? Can a rational being entertain any emotions at all toward an electronic machine?

You're darned right! Just ask anyone in the midst of a problematic impasse, known esoterically as a "bug." I soon found that the emotional response was not limited to those in the introductory stages of BASIC, because knowledge of BASIC in itself was not where the difficulty lay.

What was overlooked, while perfecting my technique of using BASIC, was the old axiom of master and slave. Mastering BASIC, I discovered, was not simply learning the slave's language, but also learning the psychology of compatible relations with one's speedy slave.

Besides the psychology needed, there was also the politics of the situation. There was an authoritative stance to be maintained, such as when king and subjects interact to carry out the rules of his monarchy. For example, if the king sets forth a decree and one of his subjects refuses to obey, it might be off with the head, into the dungeon, or banishment from the land. In like manner, when any regal programmer gets such offending disobedience as an error message, coupled with refusal to proceed, from his nasty serf, is not that the time to take hatchet in hand to said computer, sentence it to the closet, or banish it from the premises?

Absolutely! But it can be so much trouble to be kingly. Alas, one would have to garner another willing subject and buy another computer. So one lesson I grudgingly realized (which is never stated in the manual), is that genius should never be confused with royalty.

Genius does imply superiority over imbecility, so what was my trouble? I was determined to find out. I attempted a philosophical analysis. I got out my old logic books and tangled with various truth theorems. This eventually evolved into an essential question: If man is superior to machine, am I not the master and it the slave? Am I not the genius and it the idiot?

Surely it was so. After phrasing such a brilliant question, I was becoming uncertain, if not confused. I carefully and thoroughly reexamined the above question, finding nothing. Then I went back to the original premise: Man is a slow genius, the computer a fast idiot.

That's when it hit me: Slow versus fast, rather than genius versus idiot, were the words that I should have emphasized! I pondered this sudden revelation. It was on this point that I needed to focus. Continuing with this line of thought, I reasoned that the king's subjects always responded immediately, and unthinkingly. They did not take the time, nor was it necessary for them, to think through the command. They just obeyed — fast! I saw that the idea of speed was at the heart of the problem.

This concept included not only mechanical and electronic speed, but also psychological and intellectual swiftness. The programmer must go slowly, is (of necessity) slowed down by the requirements of accuracy, and works best in that mode.

I, the precocious programmer, had failed to assess my talents correctly. It is with profound concentration, and a slowing of activity, that such sequential steps as the mental working out of a FOR . . . NEXT loop, or the phrasing of a command in its proper syntax, are learned and mastered. I had observed mere repetition in that loop, had skipped over its process, and moved on to the next routine.

I had assumed the attitude of one who issues many orders to, expects much from, but all the while discounts, his slave. Hence, the slip that brought my program to a halt. My failure was in not adequately thinking out the step-by-step execution of that simple BASIC routine. I had viewed the mundane procedure, and the repetitiousness that was required to get from point one to point

two, as less than important. I, the mental giant, was brought down by a small detail!

As a consequence, I felt like a dunce rather than a genius. Such a thought was intolerable to me. I could not endure it. The contradiction became the point of my downfall, and I began to go berserk.

Could I be so stupid? I, the one with the brains? I felt strongly that I must correct this situation. I must take control and reassert my authority — establish my position. The relationship of master to slave, king to serf, genius to idiot, must be definitely understood. I must prevent further catastrophe by protecting myself from that monster machine. I, myself, must become "idiot proof." My mind was going wild with ideas on how to proceed, until eventually a short-circuit occurred which made it perfectly clear what to do. Only idiots would harbor bugs, so I devised a plan to trap them. I called it the ONERRORGOTOHXXX routine.

It worked like this . . . When my slave becomes infested with bugs and quits on me, I now slow down, count to ten, get my magnifying glass out and start examining the listing in search of those impertinent satanic creatures. When I find one of the offensive demons, one of the "bugs," I let the idiot have the full vent of my righteousness, rage and indignation. I raise my voice to shout the ONERRORGOTO message, all the while stomping my feet, waving my arms and jumping up and down. I threaten that defiant plastic box right down to the very last byte of its near-trembling baud!

Meanwhile, a FOR . . . NEXT loop is used to time this spectacular sequence. Then, the RESUME command is invoked. I regain my control through a subroutine to RESTORE clarity before the RETURN that sets me to the correct speed of a genius. I finally STOP, and the idiot computer never knew what hit it.

We are ready to start again in the proper relationship, at the proper pace, knowing full well the enlightened meaning of "Man is a slow genius, the computer a fast idiot."

Reviews

NEWBASIC 2.0
Modular Software Associates
209 18th Street
Huntington Beach, CA 92648
(714) 960-6668
$39.95 plus $2.50 shipping

NEWBASIC 2.0 adds a set of powerful commands to Disk BASIC for the TRS-80 Model I and Model III. A total of 49 new commands may be added. The total number of new commands is left up to the user in that a program called Creator is executed to generate NEWBASIC. Creator queries the user for each command prior to generating a working copy of NEWBASIC.

Some very useful graphics commands are incorporated into NEWBASIC. They include CIRCLE, DRAW, GLOAD, GPOINT, GSAVE, INV, LINE, PAINT, PGET, PLOT, PPUT, ZGET AND ZPUT. The authors of NEWBASIC have tried to use the syntax of the Radio Shack Color Computer where they could in the graphics commands. Each command has its own set of parameters. As an example, DRAW can use one or more of the following arguments: (D) draw line down, (E) draw line at 45-degree angle, (F) draw line at 135-degree angle, (G) draw line at 225-degree angle, (H) draw line at 315-degree angle, (L) draw line left, (M) draw line from current point to new point, (R) draw line right, (U) draw line up, (B) blank (i.e., move current graphics but don't draw), (N) no update (draw, but don't update current position), (Z) zero (erase line), (A) current drawing angle, (S) current drawing scale, (X) execute a string as one or more DRAW arguments.

Similar, but less numerous, parameters are used with the other graphics commands. NEWBASIC, in addition to generating extremely fast graphics, allows graphics screens to be saved as disk files and reloaded. In addition, graphics may be called into memory from disk and stored in memory, and moved to video memory for extremely fast animation effects.

A pair of sound commands are available which allow not only musical notes, but some rather interesting sound effects. The documentation lists approximate arguments that correspond to a three and one-half octave musical range.

Two types of typing shorthand features are a part of the program. QUICKEY allows typing 39 of the common BASIC keywords with a single keystroke. DEFKEY allows the user to configure 10 keys to produce an entire phrase using only a single keystroke. Using this feature, it is possible to enter an entire command such as ' CMD"DIR" (RETURN)' with a single keystroke. In addition, the defined values for DEFKEY may be saved as a disk file.

NEWBASIC 2.0 provides for some advanced programming features that speed up programming execution and make programs much easier to read and debug. CALL will execute any machine language program stored in memory. DRUN will exit BASIC and execute a disk program: DRUN"SCRIPSIT" exits BASIC and loads Scripsit.

One or the more clever commands is the DPEEK and DEPOKE pair. These allow you to peek and poke 16-bit values directly. This eliminates the confusion that often occurs in trying to remember which comes first, the lsb or msb. Both of these commands allow for decimal or hexadecimal, or computed variable arguments.

A DO . . . UNTIL command is included as well as additional forms of the conventional GOSUB and GOTO commands. An extended debugging feature is provided with an expanded Trace-type command.

My favorite command is XSTR$, which will allow a string to be executed as if it were a BASIC statement. This allows you to use LINEINPUT to enter a complex math function directly from the keyboard for evaluation by your program.

NEWBASIC is supplied on a 35-track, single-density disk. The documentation accompanying the disk gives detailed directions on how to transfer the programs to a Model I or Model III running TRSDOS, LDOS, NEWDOS80 or DOSPLUS. A 70-page manual is supplied with the software. Each command is completely explained and at least one example is given for each command. Some nifty demo programs are also included which should give the user some interesting programming examples to run and list.

NEWBASIC is a very useful tool for anyone programming in BASIC who wants very high speed graphics and animation effects that are simple to program, and needs the advanced commands and time-saving features this program provides. At $39.95, NEWBASIC 2.0 offers a lot for the money.

Harry Avant

LNW80 Model II
LNW Research Corporation
2620 Walnut
Tustin, CA 92680
(714) 544-5744
$1995

My experience with microcomputers goes back not quite five years and is strictly that of a hobbyist/ user. That is to say, I have no technical background in electronics, and no computer science degree. I have taken a few computer courses, but most of what I know is either self-taught or picked up at various local users groups. I've had my LNW for about four months and before that I had a TRS-80 Model I. I began using the TRS-80 as a learning tool and eventually started making use of VisiCalc, Scripsit, etc., for business purposes.

I'd like to say a few words about the LNW for those of you not familiar with it.

I have been interested in the LNW since first seeing ads for it in some of the national computer magazines several years ago. It has been billed as an upgrade from the TRS-80 Model I. Among the more intriguing features claimed are high resolution graphics in both black and white and color, a Z-80A running at 4 MHz, the capability to display text in an 80-column by 24-line format, and 100% software compatibility with the TRS-80 Model I.

So, after four years of upgrading my TRS-80 Model I (starting with 4K Level I and ending with 48K Level II, two disk drives, MX-80 modem, Percom doubler, lowercase, 50% CPU speedup, etc.), I decided to take the plunge and got an LNW. My LNW is a factory-built Model II, so while some of what I say will apply to both LNW models, the review is really of the Model II.

The LNW comes in an attractive, light-colored metal case which includes the whole system, excluding disk drives and monitor. In other words, it has everything in one unit that the TRS-80 Model I would have in the keyboard and expansion interface (plus some extras).

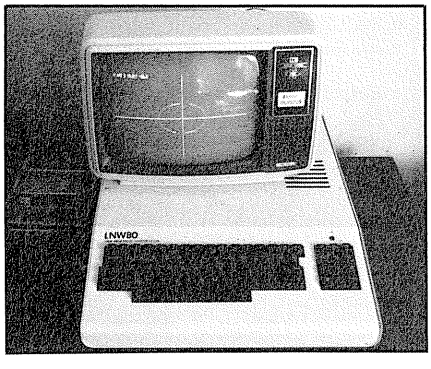
Several points should be noted regarding the keyboard. It is a 73-key typewriter-style keyboard including an 11-key numeric pad, control, shift lock, caps lock, underline, and a high/low key controlling the CPU speed. There are two reset keys which, when depressed simultaneously, cause a system reset. The control key apparently needs special software drivers as it does not normally change the ASCII value of other keys when depressed together with them. The shift downarrow provides a control key as it does on the TRS-80 Model I. There is a bad problem with key bounce on my LNW though I rarely experience it, as most of what I do is under a DOS with its own debounce routine. One other feature of the keyboard will be seen as a blessing by some and a curse by others. Unlike the TRS-80 Model I, the LNW arrow keys are all together on the right-hand side of the keyboard. For some business applications, this will simplify things, while for other uses (games in particular) it can be a problem.

There are three edge-card connectors on the back of the unit, a floppy disk connector, a parallel printer connector, and an expansion bus connector. The disk and printer connectors plug in upside down as compared to the TRS-80 Model I. The expansion bus connector plugs in the same as the TRS-80 Model I. The floppy disk controller is an LNW 5/8 doubler which will support 5¼" and 8" dual- or single-headed floppy disk drives in either single- or double-density.

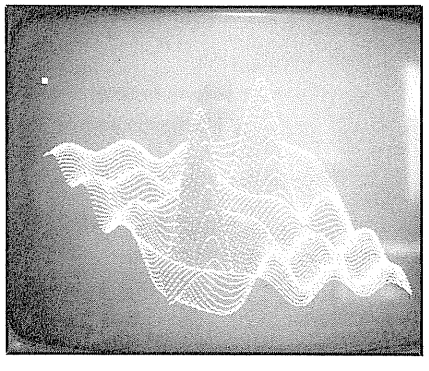
The LNW Model II comes with 96K of user RAM, 16K graphics RAM, and 16K of ROM, CRT memory, keyboard addressing, etc. This, however, can be confusing and misleading. Since the Z-80A can only directly address 64K of memory and 16K of that is normally the ROM, special software is needed to make use of the extra memory. Such software would "page" or "bank select" the extra memory and while it is not very difficult to implement, it is by no means trivial. As of this writing, very little software is available which does the paging. LNW does have a version of CP/M 2.2 which switches out the ROM and lets you run "regular" CP/M, but I have no CP/M software other than the operating system itself, so I don't have any firsthand experience with it.

The graphics RAM is independent of the regular CRT memory, so high resolution graphics and normal text can be on the screen simultaneously, even overlapping. Four different graphics modes are possible. In normal TRS-80 mode, the high-resolution graphics RAM is not used at all. Next, there is a low-resolution color mode. In this mode, the user has control of 128 x 192 pixels with eight colors. Using this mode, a user can utilize a color TV with RF modulator as a monitor and get quite satisfactory results. There is also a high-resolution

The LNW80 Model II with black and white monitor as it is used in Mr. Irwin's home.



A sample of the high-resolution graphics that are possible on the LNW80 Model II.



black and white mode. In this mode, you have control of 480 x 192 pixels. Excellent high-resolution graphics can be accomplished with it. Sine waves, circles and drawings of all kinds are easily done (with proper software, of course). This mode is used to generate the 80x24 text mode mentioned in the LNW ads. This, again, is somewhat misleading in that special software drivers are needed, and none of the software drivers I have seen are compatible with all software. Most conspicuously, none will yet allow 80 x 24 text with Scripsit or any of the other major word processors. Lastly, there is a high-resolution color mode. Here, you have control of 480 x 192 pixels and eight colors. This mode is the least software-supported of the graphics modes. To use this mode, it is necessary to have a high-resolution RGB monitor.

The LNW is capable of using a black and white monitor, a composite color monitor, or a high-resolution RGB monitor. The black and white and composite color monitors both connect using an RCA-type jack. The RGB jack is a six-pin DIN-type female connector. All three jacks are provided and three monitors could be used simultaneously.

The LNW has several other nice features as well. It has a cassette DIN connector and an RS-232C interface, but it uses a DB25 female connector rather than an edge-card connector. The Z-80A will run at 4.0 MHz, however certain I/O situations require a slower processing cycle. I encounter this most often in booting old TRSDOS diskettes. They will not boot at the higher speed. There are two ways to overcome this. There is the high/low key on the keyboard which toggles the CPU between 4.0 MHz and the TRS-80 Model I speed of 1.77 MHz. The other possibility is a switch on the back of the unit. Turning this switch on will slow the CPU down to 1.77 MHz whenever disk I/O is taking place. One final feature they added is the quiet, built-in fan.

Another of the selling points in the LNW ads is the 100% software compatibility with the TRS-80 Model I and, in the case of the LNW Model II, the CP/M compatibility. I have no experience with the CP/M software, but do know that such software is somewhat more difficult to transfer from machine to machine than some ads would have you believe. As to the TRS-80 software, that is one of the major reasons I decided for LNW as opposed to another brand altogether. I have not been disappointed. I have yet to find any of my software which would not run. Some people have reported problems with programs using joysticks, but I have had no problem with my Alpha Joystick.

In addition to the TRS-80 software, there is a growing body of LNW-specific software. "LNW specific" means that either high-resolution or color graphics are used. Perhaps the most useful program for LNW owners is LNWBASIC from Modular Software Associates. This enhanced BASIC provides the LNW owner with a very good tool for using the high-resolution graphics. It has such commands as circle, line, draw, color, pset, preset, etc. In addition, it has some non-graphics-related enhancements, such as call (address), Do . . . Until (condition), sound, spoolon, spooloff, etc. One other feature which may help in software development using LNWBASIC is that it has been made syntactically quite similar to TRS Extended Color BASIC. There are a few small bugs in the version I received and there is little support for the high-resolution color graphics mode, but I understand an upgraded version is in the works.

Another graphics package available now is the HIRES 1 & 3 software from E & H Software. The HIRES software gives a limited 80-column ability, turtle graphics and screen dump. However, it is much faster in execution than LNW-BASIC.

Another useful product is the DHARM character generator by Balaclava Software. This program allows the user to define his or her own character set. Include foreign, scientific, game or other characters and use them in BASIC programs as well as some machine language programs. No compatibility, as yet, exists with Scripsit.

Several screen dump programs exist which allow the user to dump high resolution screens to printers with bit-graphics capabilities. Also, several programs are now available which permit plotting of graphs, charts, etc., both on the screen and to printers. As far as I know, there is only one color game written for the LNW (in low-resolution color). It is Laser Command from the Maine Software library. It is not as sophisticated as some of the newer TRS-80 games, but it is well done and is a first step toward better things.

I am very satisfied with my purchase. The 4MHz speed saved me several days inputting data into a ledger program using VisiCalc. I have heard several complaints about LNW's unresponsiveness to questions or problems, but most

of these have been from people who built their LNW Model I from kits. My unit was factory-built and the only occasion I had to contact LNW was to request a copy of a newly-published user's manual. It was promptly forwarded to me. Other than a few minor complaints, I have nothing but praise for what is a well thought-out, well-built machine.

David Irwin

Ed. note: The LNW 80 Model II is only available in completed form. The LNW 80 Model II, not including a video monitor or disk drive, sells for $1995. The LNW 80 Model I is still available in kit form. Several options are possible and readers should contact the company for complete details. The addresses and phone numbers of the companies mentioned in this article are:

LNW Research Corporation, 2620 Walnut, Tustin, CA 92680, (714) 544-5744.

Modular Software Associates, 209 18th Street, Huntington Beach, CA 92648, (714) 960-6668.

E & H Software, 11814 Coursey Blvd, Suite 249, Baton Rouge, LA 70816, (504) 293-3400.

Balaclava Software, P.O. Box 281, Altadena, CA 91001, (213) 794-4796.

The Maine Software Library, P. O. Box 197, Standish, ME 04084.

Several bulletin boards and newsletters have begun for the LNW 80 owner. You may wish to contact them for further information.

LNW User (newsletter), 4345 Manchester Road, Grand Island, NE 68801.

LNW News, 244 Mill Road, Yaphank, NY 11980.

LNW BBS of the Southwest, Baton Rouge, LA, (504) 291-4331.

LNW BBS by Inflo, Inc., Yaphank, NY, (516) 924-8115.

SuperScripsit 1.1.0
Radio Shack Stores and Computer Centers
Catalog #26-1590
Models I/III Disk Systems $199

Radio Shack has released an updated version of SuperScripsit, version 1.1.0, which corrects some of the problems associated with the original release. A six-page addendum to the documentation adds clarification for some commands and provides a very necessary warning when using the block move command.

With the original release, moving a large block of text could often prove disastrous if insufficient disk space was available. The addendum warns you to check the amount of free space available on drive zero for the file MOVE/CTL prior to editing a document. By following this advice, very large blocks of data can safely be moved without losing your file.

An upper limit for document size has been established at slightly over 170 thousand bytes. This restriction applies both to floppy and hard disk systems. The limit seems to have been determined to allow for floppy backup from hard disk files. Since the file size limit corresponds to about 45 pages of very tightly typed text, this seems like a reasonable restriction. Version 1.1.0 now includes a DO file for patching the program for use with a hard disk and the LDOS operating system.

It appears that this updated release has not lost any of the special features that made the original version so powerful. The on-screen help feature is still present as well as the ability to merge form letters from Profile III. The special printing capabilities are more accessible now that drivers for all of the current Radio Shack printers are provided.

The addendum suggests that unnecessary drivers be eliminated from the working disk. I would suggest that you kill any file on drive zero that is not going to be used. This is especially important if you are going to keep MOVE/CTL on drive zero.

One noticeable omission in SuperScripsit is the absence of a chain command that will allow you to bring in files or stock paragraphs. In many applications, the same paragraphs or even pages of text are used over and over. I'm sure that business users will notice this shortcoming. Perhaps, someday, there will be an upgrade that adds this feature.

Another limit is that only two different headers and footers are allowed per document. With a text capability of at least forty-five pages, the program should provide for more changes in headers and footers.

Compared to the original Scripsit, SuperScripsit is quite an improvement. With Scripsit, text was limited to memory size, or about twenty-two pages, but now document size has been increased by a factor of at least two. Scripsit always left me guessing as to where page breaks would occur. SuperScripsit, with its line- and page-counter display at the bottom of the display means that you can determine page breaks before printout. With the horizontal scrolling feature, it is quite easy to visualize how text will appear on paper. The best features of SuperScripsit, as compared to Scripsit, are the printer controls that are available such as super- and sub-scripting, underlining and bold face printing.

Harry Avant

The VC Formula Printer
By John Campbell & Ken Edwards
Dist. by The Alternate Source
704 North Pennsylvania Ave.
Lansing, MI 48906
Model I/III $39.95
Model II version $49.94 under special order

Like many serious VisiCalc users, I have been frustrated by the method that VisiCalc uses when it prints out hardcopy listings of the cell position formulas in reverse order and in a long "laundry list". The authors of this utility have provided a welcome method that allows us to print out the formulas in a layout that humans can follow.

They have accomplished this by writing a series of linked machine language programs that read a standard VisiCalc file and then send to your printer the formulas in a layout that is like the spreadsheet you created. The program can also route the spreadsheet information to a file that can be read by your word processing program.

The programs are supplied on a double-sided diskette which has the Model I version on one side and the Model III version on the other. Complete, commented source code is available for an additional $50. The programs require 48K of memory. To use the programs you first copy the files from the master diskette to a system diskette that contains no other files. Most major operating systems are supported, including TRSDOS 2.3 and 1.3, DOSPLUS 3.4, LDOS 5.1, and NEWDOS/80 v2.0. My evaluation was done using TRSDOS 1.3 on a Model III, so I cannot comment on the program's use on other systems. It is possible to run the programs with a one disk system, but two are required if you are using Model I 35-track drives.

You execute a DO file which loads the programs and they present you with a series of prompts. The prompts ask whether you want to select a printed report or a disk file, if you want to print the global specifications, and how to format the output (page length, line length, column spacing, and characters per column). Adequate default values are provided and there are helpful author's comments in the manual. Once the prompts are answered you have a chance to correct your input and select printer control codes. The programs then read the VisiCalc file and produce the desired output. Once the processing is complete you can start over or quit.

Needed Enhancements

In my testing for this review there were three features that I hope future releases of this product will contain. All of the printing is performed at once and there is no time to insert additional pages in a single sheet printer. There should be the option to halt between pages. I did get around this annoyance by storing the formulas in a text file, loading them into Scriplus, and selecting the pause between pages option. However that requires that the available disk storage space be at least four times the size of the VisiCalc file being processed.

A second enhancement should allow you to reprocess the spreadsheet without having to go through the complete prompt process again. I found that I would like to make multiple copies of spreadsheets, but I had to start over at the beginning on each one.

Lastly, while the 26 pages of documentation appear complete (and include a sample lesson), they could be made simpler. They should place run instructions right at the beginning, not page 11, and reference the appendix for the various operating systems. In addition, the instructions for TRSDOS 1.3 as printed are incorrect, but the correct instructions could be found in Appendix B. It took me several minutes to figure that out.

Recommendation

This product provides an excellent, easy-to-read, logically arranged listing of the VisiCalc formulas. The programs include the possibility of printing out spreadsheets without any data included so your confidential data can be protected. It solves for many of us the problem of obtaining easily read and understandable hardcopy backup. It aids in program development by eliminating the wasteful use of computer memory due to unneeded cell positions that hold data of labels that are too long.

The Alternate Source also sells Spreadsheet Application Planner Pads for $7.95. But, I believe that you would do better to spend $39.95 for the VC Formula Printer and handwrite your spreadsheets. The programs belong on every serious VisiCalc user's diskette.

Timothy K. Bowman

VIS\Bridge/SORT
Models I/II/III/12/16
$89
Solutions, Inc.
Box 989
Montpelier, VT 05602
(802) 229-0368

This VisiCalc utility allows you to sort your spreadsheet files. It is one of a series of VisiCalc enhancements called VIS Bridge that Solutions Inc. has released for the TRS-80 computer line. The SORT utility lets you specify up to five sort levels within one spreadsheet. Sorting can be on alphabetic or numeric data, in ascending or descending order.

We tested the product on a Model II system and found it easy to install and implement. The utility comes with four program files and a demonstration file. To sort an existing spreadsheet, you first load it from disk, insert a row or column and save it back out. To specify the row or column to sort on, its priority in the sort, and its direction (ascending or descending) you insert easy and obvious commands in the row (or column) you added to the spreadsheet. Then save the data you want sorted out to disk as a new file under the DIF format. Now you leave VisiCalc and run the BASIC VBSORT program. It will ask for the DIF filename, type of sort to perform, and the row (or column) where you specified the sort parameters. This will create a new file to be merged with your existing spreadsheet.

This part is quite clever. You have just created a sequence of VisiCalc move instructions which, when merged with your existing spreadsheet, will put the data in sorted order. The process is done. Now the altered data can be saved back to disk.

It was simple and easy to do. The

Get to know ...

MR. SOFTY

the affordable GAME SOFTWARE *MOD. I, III DOS compatible*

10 GAMES-ONE PACKAGE -on one cassette-

Requires 16K RAM-up. ONLY $10.95

Shooting Gallery	Purple Puma
Needles in the Hay	Concentrate
Math Smarts	Word Works
Dodge It	Space Races
Hop Along Cass-80	X's and O's

Disk Version Coming.
Look for second package soon!

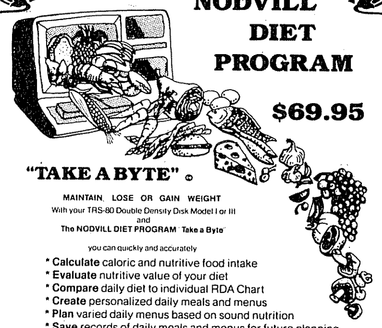
Please include $1.50 for S/H.
SEND CHECK OR MONEY ORDER TO:

MR. SOFTY

1416 Ridgewood, Lakewood, Ohio 44107

OHIO RESIDENTS ADD 6½% SALES TAX

NODVILL DIET PROGRAM



$69.95

"TAKE A BYTE" ©

MAINTAIN, LOSE OR GAIN WEIGHT
With your TRS-80 Double Density Disk Model I or III
and
The NODVILL DIET PROGRAM "Take a Byte"

you can quickly and accurately

- * **Calculate** caloric and nutritive food intake
- * **Evaluate** nutritive value of your diet
- * **Compare** daily diet to individual RDA Chart
- * **Create** personalized daily meals and menus
- * **Plan** varied daily menus based on sound nutrition
- * **Save** records of daily meals and menus for future planning
- * **Print** nutrition charts, food, meal, menu, and grocery lists

"**Take a Byte**" is a Modular **BASIC** Program

MAIN Program Menu
(1) Recommended Daily Dietary Allowance (RDA) CHART
(2) 733 Expandable Random Access FOOD LIST Data File
(3) 28 Nutritional MEAL LIST Data File Examples
(4) 7 Balanced DAILY MENU LIST Data File Examples
(5) GROCERY LIST Program Module
(6) 25 Page USER'S MANUAL including Charts and Tables

ALL Data Files can be EXPANDED and MODIFIED systematically and flexibly to reflect your personal diet, your choice of diet book, or your doctor's suggested diet.

For more information: Call: (203) 431-6449
To Order: Mail a check or money order for $69.95 to:
NODVILL Software
24 Nod Road
Ridgefield, Conn 06877

TRS-80 is a Trademark of the Tandy Corporation

Color Computer Secrets Revealed

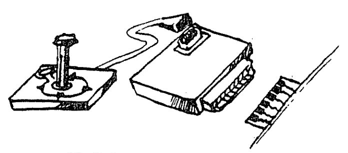
Learn all about the secret inner workings of your Color Computer. This new book tells you how to:

- Make back-up copies of machine language programs.
- Merge two or more BASIC programs.
- Increase your memory for free.
- Upgrade your Color Computer to 16K, 32K or 64K RAM and add Extended BASIC.
- Run your computer at twice normal speed.
- Use a myriad of PEEK and POKE statements . . . and much, much more.

All this in an easy-to-read and profusely illustrated book. A must for every Color Computer owner — just **$11.95** + **$2** shipping. Order direct from:

DISK 'N DATA
5450 Rugby Street
Burnaby, B.C., Canada V5E 2N1

THE JOYSTICK INTERFACE



USE THE JOYSTICK YOU HAVE!!

Simply plug your joystick into THE JOYSTICK INTERFACE, and plug it into your Model I or Expansion Interface.

1. Compatible with most major games.
2. Low cost.
3. Use your present or favorite joystick.

ONLY $24.95!!

FREE SHIPPING

ATARI (TM) JOYSTICK $6.50

The Software Exchange
1615 Compton Road
Cleveland Hts., Ohio 44118

Ohio residents add 6.25% sales tax.
Model 3 & 4 available soon * Dealers invited

LIMITED OFFER

NEWDOS - 80 ver. 2.0
and DOSPLUS ver. 3.4

A combined regular price of $299.96

Now $149.00 combined price

or call for separate pricing. Offer subject to cancellation without notice.

CALL TOLL FREE
800-835-0071

In Kansas 316-665-3611

SALES DATA, INC.

526 E. 4th

HUTCHINSON, KANSAS 67501

The House-Ware Genies

- ★ The Mail Genie
- ★ The Pantry Genie
- ★ The Recipe Genie
- ★ The Calendar Genie
- ★ The Insurance Genie

At last, the Turn-Key Programs that allow Dad to influence Mom, that a TRS™-80 Model III and Color Computer is a modern day necessity.

Introductory Price **$29.95**

Developed By
SOUTHERN CENTER
FOR RESEARCH & INNOVATION
P.O. Box 1713
Hattiesburg, MS 39403
TELEPHONE 601-545-1680
TELEX 585400 SCRI US HATI
CABLE SCRI US

Dealer Inquiries Invited

LARGECAPACITYSYSTEMLARGE

SMALL & reg. 32K 2 drives

LARGE CAPACITY PROGRAMS

M1 M3

ACCOUNTS RECEIVABLE
5000+ ACCOUNTS

30-60-90 DAYS AGED STATEMENTS SHOW DATE/INV#/DESCRIP/AMT (WITH AGEING). SELECTIVE FINANCE CHARGES & RATES, FAST ENTRY, POSTING W/AUDIT REPORT, SUB-ACCTS, % OF CREDIT LIMIT, DATE OF LAST PAYMENT, SALES ANALYSIS.

GENERAL LEDGER
400+ ACCOUNTS

+REPORT FLEXIBILITY/CAPACITY
+DEPARTMENT P&L (UP TO 9)
+UNLIMITED ACCOUNT CATEGORIES
+STATEMENT OF CHANGES (ASSETS ONLY)
+PERCENT P&L

149.95 Each ★ Both for 199.95

HOLMAN D-P SERVICE
2059 WEST LINCOLN
OROVILLE, CA 95965
916-533-5992

3.00 S&H
VISA OR MC
COD

manual $30.00 test set $50.00

Get

FREE

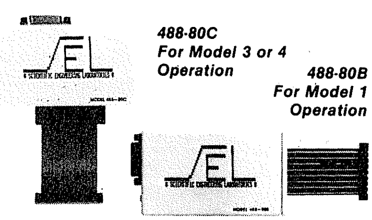
Programs for the
COLOR COMPUTER
Join the

International Color Computer Club, Inc.

Main Office
2101 East Main St.
Henderson, TX 75652

IEEE-488 TO TRS-80* INTERFACE

Everything needed to add *powerful* BASIC GPIB-488 controller capability to TRS-80 Model 1, 3 or 4, Level 2 or DOS with a minimum of 16K.



488-80C For Model 3 or 4 Operation

488-80B For Model 1 Operation

Model 488-80B or 488-80C Price: $375
+ shipping, insurance & tax

WHEN ORDERING SPECIFY DISK OR TAPE

SCIENTIFIC ENGINEERING LABORATORIES

11 Neil Drive • Old Bethpage, NY 11804
Telephone: (516) 694-3370

**Trademark of Tandy Corp.*

There is no affiliation between Scientific Engineering Laboratories and Tandy Corp. or Radio Shack.

utility was efficient and sorted quickly. You do have to watch out for subtotals or other entries which you may not want sorted. You do have to be careful not to confuse your filenames. It would have been nice if the utility would kill off the intermediate files that are not needed after it is done.

The documentation is adequate. There is one glaring omission. Purchasers are given 45 days of free telephone support but nowhere do they list the number to call. The utility is easily backed up and a straight-forward demonstration session is included for first-time users. It works well, but only heavy users of VisiCalc will have reason to invest almost $90 in a sorting routine. For many users, the VisiCalc move command, and some careful design when starting a spreadsheet, is sufficient to keep the data in a reasonable order.

Besides the SORT utility, Solutions Inc. also has offers a Report, a Dow Jones, and a General Ledger (Model II) enhancement for your VisiCalc.

Cameron C. Brown

Quick Sort III and
System Tape Copy
DCS Software
86 Mansel Drive
Landing, NJ 07850
Quick Sort, Models I/III, $8.95
System Tape Copy, Models I/III, $5.95

For tape-based TRS-80 users, DCS Software has recently introduced two new software utilities — Sort and Copy. The utilities are inexpensive and add flexibility to your computer.

Copy is a machine language utility for copying machine language programs. The program works on Model I or III and it allows speed changes for the Model III. For example, you've just received a great new 48K adventure recorded at 500 baud; with Copy you can load the program in at 500 and save it back out at 1500 baud. This cuts your loading time by two-thirds. Very nice for those long programs.

The program is useful for making backup copies of your valuable machine language tapes. Any number of copies can be made at one time with Copy. The program is easy to use and functions perfectly. Anyone who uses a great many system-formatted tapes should consider this utility.

Quick Sort III adds a new command (SORT) to your BASIC interpreter. This command allows you to sort a string array in either ascending or descending order. The program is a machine language utility that makes sorting much faster than similar coding done in BASIC. Either single- or multi-level arrays may be used.

Using the SORT command is very simple. The syntax is SORT A$(A,2) where A$ is the string to be sorted. A indicates ascending and 2 tells the routine to order the second level of the array (i.e., A$(0,2), A$(1,2), A$(2,2) . . .).

Sorting time for a 100-element array is three seconds, 250 elements is fourteen seconds, and 500 elements is sixty seconds. The times seem to grow logarithmically, rather than linearly, as with most sorting programs. Certainly, it's not the fastest sorting program on the market, but it's the easiest to use and the least expensive.

For the beginning computerist, these two programs could be the start of a library of useful utilities. It's nice to see that even after five years, TRS-80 users can still buy quality software for even the smallest systems.

Mark E. Renne

101 Color Computer Programming Tips & Tricks
By Ron Clark
ARCsoft Publishers
Woodsboro, MD 21798
$7.95 paperback

Any Color Computer owner can appreciate my excitement when I saw the advertisement for *101 Color Computer Programming Tips & Tricks*. I couldn't get my money in the mail fast enough.

The book arrived within a couple of weeks and I was anxious to dig into the "hints, secrets, shortcuts, and color techniques" that the book promises. It didn't take long for disappointment to set in. I had expected a book crammed with the esoteric little tid-bits that we all devour magazine articles for. Instead, I got a book that is about one-third fluff or useless. Much of the rest is ordinary.

The book does contain 101 short listings and narrative explanations, but several of them seem to be nothing more than modifications or slight variations on examples from Radio Shack's *Getting Started with Color BASIC* or *Going Ahead with Extended Color BASIC*. Some of the listings are short excerpts of code from larger listings that are elsewhere in the book. If you assume that anyone reading a book on programming tips and tricks knows the basics of BASIC, this book is practically useless to him.

Equally disappointing are the numerous misprints and outright errors in the book. The most glaring examples are tips 80 and 81. These two tips attempt to show the effects of the FIX and INT statements (are these tips & tricks or tutorials?). Neither of the tips show anything because the random number generator is improperly used in the example. At the end of this review are a list of some of the errors and misprints I discovered. There may be others, but it should help those of you who already purchased the book.

The section on sound and music is the poorest. I found no tips or hints of any real value. I had hoped for some tips on how to translate sheet music or create realistic sound effects. The two tips that use the PLAY statement generate a ragtime piece and a minuet. They were enjoyable.

It is not a complete waste of money. The book does contain some tips of value. The section on color graphics is interesting. There is a good illustration on the use, and limitations, of PPOINT. One tip cleverly uses a digital clock for demonstration. Some interesting, but primitive, animation is shown in two other tips. The scale parameter of the DRAW statement is used to create a feeling of perspective. The section concludes with several graphing tips and how-to-do-it information. Nothing was included on the PUT and GET statements.

Text techniques are well covered. A good tip is given to repeatedly print a character during input through using the INKEY$ function. Another tip does a fancy job with the PRINT USING statement.

The section on fun and games didn't have much of either. It does, however, have a nice dice rolling routine that uses text screen graphics.

I call it fluff but the novice programmer may appreciate the elementary slant of the game routines. I think most readers will find something of interest in the book. Until more Color Computer books are available, it does have a place in your library.

Eratta

1. In tip 9, delete lines 50, 60, 70, and 80. Change line 40 to be: 40 LINE (0,0)-(255,191), PSET, BF.
2. In tip 11, line 90 should be: 90 LINE (128,80)-(128,160), PSET 'TORSO.
3. In tip 12 the circle is too large to paint around. Change line 40 to read: 40 FOR L=20 TO 80 STEP 20. Change line 70 to be: 70 FOR L=100 TO 160 STEP 20.
4. In tip 25 lines 70 through 100 have the color parameters in the wrong order. They should be in the order C1, C0, C1, and C0.
5. In tip 29 delete the last two characters, ",6".
6. In tips 80 and 81 change lines 10 to be: 10 R = RND(100)/10.
7. In tip 85, the number 16 in the first sentence of the narrative should be 9.
8. In tip 100, the comma just before the variable P$ in line 20 should be a semi-colon.

Stephen G. Stone, III

COPYCAT
DISK BACKUP UTILITY

- Allows you to duplicate most protected software
- Automatically copies: **$34.95**
  - Multiple Sector Sizes
  - Dual Density Tracks
  - Any Sector Numbers
  - False Track Numbers
  - Bad Sector ID's as used for Protection
  - False Side Numbers
- Complete error messages
- Requires 48K and 2 disk drives
- Specify Model I or Model III

OMNI SOFT RESEARCH

2170 W. Broadway, #501 B
Anaheim, CA 92804
714-772-5000

V I Z . A . C O N
A CONSOLIDATION SYSTEM for VISICALC users

New product adds 3-dimensional capability to any VISICALC model

Now with the help of VIZ A CON you can combine multiple "pages" of data from a model for heirarchical consolidations (eg. Dept., Div., Co.) or for summations over periods of time (eg. Week, Month, Quarter, Year-to-date).

Typical uses are to combine weekly sales reports or departmental budget data. You can create a complete network of consolidation processes and modify it any time (eg. for Merger & Acquisition analysis).

Special formulas (eg., ratios, percentages) can be recalculated after any consolidation. VISICALC precision is maintained for all data. You can customize titles, row and column headings, footnotes, etc. for each report.

VIZ A CON creates data files usable with VISICALC. After VISICALC "what if" games, use VIZ A CON to find out what happened.

Model I/III **$89.95** — Model II **$119.95**
TOLL FREE 24-HR. SERVICE **800-547-5995** (Ext. 170)
— Visa/Master Card —

Or mail Check to
A B A C U S ASSOCIATES
Suite #240, Dept. 101
6565 W. Loop South, Bellaire, TX 77401

"Creating Simple Solutions to Complex Problems"

We'll give you this $175 software package for $24

2 Games! 7 Personals! Mod I/III

OSCAR DRAFT, SPECIAL MISSION PILOT
Exciting flight program. Full instrument panel. Real navigation, bombing, strafing, dogfights. Great graphics. 32K

THE WIZARD'S CITY. Adventure

PERSONAL PROGRAMS
Income Tax—Financial Statement—Mail List—Bus. Mileage —Tax File Income/Expense—Stock Charting.

DISK ONLY. IF YOU USE TAPE, ORDER DISK AND HAVE IT COPIED. AT $2.67/PROGRAM, HOW CAN YOU LOSE?

Add $1.50 shipping.

BAP$ 6011 SAN FELIPE HOUSTON, TX 77057

Do you throw away your car when it's out of gasoline?

Then why throw away your printer cartridge when it's out of ink?

We can show you how to reink your EPSON* ribbon cartridges, or other popular brands, with only a few cents worth of special ink.

Write today for **free** lists telling us your printer make and model. A postcard will do. Information available by mail only.

ZYGOTRON
P.O. Box 27
Fremont, MI 49412

*EPSON trademark Epson America, Inc

TRS-80 MODEL I T.M.*
GOLDPLUG - 80

Eliminate disk re-boots and data loss due to poor contact problems at card edge connectors. The GOLD PLUG - 80 solders to the board card edge. Use your existing cables.

CPU/keyboard to expansion interface $18.95
Expansion interface to disk, printer, RS232, screen printer (specify) . . . . . .$9.95 ea
Full set, six connectors. . . $54.95

VISA — master charge

EAP COMPANY
P.O. Box 14, Keller, TX 76248
(817) 498-4242
*TRS-80 is a trademark of Tandy Corp.

DISCOUNTS!

ELEPHANT DISKS

type	1 box	2+ bx
5" SSSD/soft	21.95	20.95
5" SSDD/soft	24.95	23.95
5" DSDD/soft	32.50	30.95
8" for Mod II	42.50	40.95
8" for Mod 16	56.95	54.95

SOFTWARE SPECIFY MOD I OR III

MULTIDOS	89.95
Z DOS	35.95
E BASIC	26.95
NEWSCRIPT 7.0	99.95
TRASHMAN/disK	31.95
REGILIAN WORM	13.50/tape 15.95/disK
GEAP/DOTWRITER 1.5	84.95
FONT DISKS 1,2,3,4	ea 25.95

HARDWARE

C.Itoh PROWRITER 1 (parallel)	479.00
PROWRITER 2 (parallel)	679.00
Printer cable (I/III or II)	31.95
HOLMES SPRINTER	89.95

Add NJ Sales Tax if applicable.
Shipping (48 States) $2.50 per order.

DIRECT-TO-TAPE
14 Station Ave. Haddon Hts. NJ 08035
Checks or cash, no credit cards.
Purchase orders add 10% to prices.
Write for catalog/newsletter

COMTRONIC SYSTEMS

TRS-80 Model I, III and Color Computers. 16K RAM Required.

ARCADE GAME	TAPE	DISK
B-52 Strategic Bomber	$10.95	$12.95
Centurion	$10.95	$12.95
F-15 Death Pilot	$10.95	$12.95
Raid on Entebbe	$10.95	$12.95
Street Fight	$10.95	$12.95
Submarine Attack	$10.95	$12.95
Air War	$ 9.95	$11.95
Killer Satellite	$ 9.95	$11.95
ADVENTURE GAMES	**TAPE**	**DISK**
Space Colony	$10.95	$12.95
Air Cav	$ 9.95	$11.95
WAR GAMES	**TAPE**	**DISK**
Battle of Midway	$10.95	$12.95
Next War	$10.95	$12.95
Nuclear Holocaust	$ 8.95	$10.95

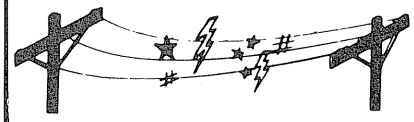
DISCOUNTS:
Buy 2 Games, Save 10%
Buy 3-4 Games, Save 15%
Buy 5 Games, Save 20%

Add .55 shipping or $2.55 for C.O.D.

"SATISFACTION GUARANTEED"

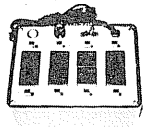
COMTRONIC SYSTEMS
4028 Somerset Lane, Kent, WA 98032

POWER LINE PROBLEMS?



SPIKE-SPIKERS™...The Solution!

Minimize equipment damaging spikes and conducted RF noise to or from sensitive equipment. Transient surge protection plus low pass RFI hash filtering. All units 120V 15A.



MINI II $44.95
Wall Mount
3-stage filter
2 sockets

QUAD II $59.95
Wall mount. Dual 3-stage filter
4 sockets & light

DELUXE $79.95
Dual 5-stage filtered ckts. 8 switch sockets main switch, fuse, light

VISA

Kalglo Electronics Co., Inc.
65 84 Ruch Rd., Dept. 80-U S
Bethlehem, PA 18017

Order Factory Direct
215-837-0700
Out of State
800-523-9685

DEALERS INVITED

PA Res. Add 6% • COD add $3.00 + Shipping

BARCLAY WHYTE ASSOCIATES

WORDPROC I-III VERSION 3-0

Basic plus machine language Word Processor

TRS-80* Model I/III 48K Disc

Special Version Now you can embed control codes in your text for

Superscript Subscript Underline Italics More

Supports all MX-80** typefonts

Still only **$14.95 each, with disc**
$10.95 each, documentation only

BARCLAY WHYTE ASSOCIATES
P.O. Box 948,
New Westminster
B.C. Canada. V3L 5C3.
Add $1.00 for postage
B.C. Residents add tax at 6%

*Trade mark of Tandy Corporation
**Trade mark of Epson America

Kraft Systems Joystick
Model KJS-01T
Color Computer Joystick
$64.95 each
Kraft Systems Co.
450 W. California Ave.
Vista, CA 92083
Western U.S.: (800) 854-1923
Eastern U.S.: (800) 633-1478
California: (800) 542-6436

Though Color Computers are inexpensive, their performance is anything but "cheap" — except for the joysticks. Critics contend that Radio Shack's joysticks are sluggish, imprecise, have poor "feel," and look "cheap." Kraft Systems, which may be the world's largest manufacturer of precision control sticks and gimbals (used in graphic display systems, medical systems, and radio controlled models), has designed a joystick to overcome these criticisms. It is easy to use, precise, never sluggish, has excellent (and adjustable) "feel," and is definitely *not* cheap.

Unlike some sticks developed for the Color Computer, no special software or adapters are necessary for the Kraft joystick. It plugs right in. Though Kraft claims it is "color matched to your computer," the stick is a true battleship gray and not a Tandy silver. The "fire" button is smaller than that on Radio Shack's stick, and is also located on the top left corner of the base, making it slightly less convenient for left-handed users.

On the bottom of the case are two recessed "mode" toggle switches. Depending on how they are set, the stick is either "free floating" (like the Radio Shack joystick) or "self-centering" (springs return the stick to the center on release). Two trim controls on top of the case allow for extremely accurate adjustment of the stick. Using these controls, the stick can be "centered" anywhere on the screen. Through a combination of trim control adjustments and mode switch settings, it is fairly easy to set the stick to act like a paddle — great for tennis and "break-out" type games.

As far as performance goes, this is as much a function of application as stick quality. For drawing detailed, accurate pictures on the screen, the Kraft joystick is clearly superior to Radio Shack's stick. In some games, such as Spectral Associate's Ghost Gobbler and Radio Shack's Polaris and Galactic Attack, players noted a 30% to 120% improvement in their scores. On the other hand, little difference was noted when playing Radio Shack's Project Nebula or Pinball. Probably the best indication of quality was revealed through the democratic process: In two-player games using both the Kraft and Radio Shack sticks, the weaker players *always* chose the Kraft joystick.

There still remains the question, "Does anyone *need* a $65 joystick?" If you are interested in game playing, probably only one is required. Few games require two joysticks, and in two-player games, the better player can always be gracious and accept the "handicap" of a Radio Shack stick. For professional game and graphics designers, play testers, and owners who want the very best, a pair might be a better investment.

Lawrence I. Charters

Rivet Race
William J. Cain
Quality Software and
Consulting, Inc.
P.O. Box 11355
Kansas City, Missouri 64112
Model I & III tape $14.95
Model I & III disk $18.95
32K — Joystick compatible

I don't know why it has taken all this time for the crazed carpenter to finally come to the TRS-80 to attempt rescue of his girlfriend from the mad ape, but he's here now from Quality Software.

The ape stands majestically on the top floor of a multi-level building with levels connected by ladders. That wild ape is tossing barrels down on our hero to prevent him from reaching the top floor and coming to the rescue of his female companion. Even reaching the top floor doesn't guarantee freedom for the little lady as the ape just takes her higher.

The second level requires the destruction of the building to beat the monkey. This is accomplished by removing six pins which hold the floors up. Seems like some bad structural engineering in this building. Of course, the ape still has another trick up his sleeve — the third level.

The last level is where I fall apart on this game as well as at the arcade. The levels are now joined together by moving elevators. These elevators require perfect timing to jump on and I never seem to quite make it across the screen.

Arrow keys control movement and the spacebar is used for jumping. Joystick operation is also supported. One or two players may participate and the game may be paused. Something missing, however, is sound. No sound effects are included at all and no provision is made for high scores.

Bringing this arcade game to the TRS-80 was a difficult task. Although the game does not have all the bells and whistles of the arcade version, it still is an acceptable substitute.

Mark E. Renne

Revenge of Rivet Race
By William J. Cain
Quality Software & Consulting
P.O. Box 11355
Kansas City, MO 64112
Model I/III Tape $14.95, Disk $18.95

The mean ol' carpenter has trapped your Daddy Ape and you must free him by unlocking his cage. You must fight off bats and monsters while maintaining your grip on perilous vines.

Revenge of Rivet Race is a sequel to another game from Quality Software & Consulting called, simply enough, Rivet Race. In that game you were the carpenter trying to save your girlfriend. This time the roles are reversed; you're the monkey trying to save Daddy Ape.

The game is a high-speed machine language arcade game, complete with sound. There are three separate screens, each with unique qualities and hazards. Each has a series of vines that you swing back and forth on while avoiding the monsters. Fruit is strewn about the vines, to be dropped on the bad guys for points. The goal on the first level is to reach the top and touch a key that unlocks pop's cage.

To surpass the second level you must push eight keys to the top of the screen to release the captive patriarch. The third level consists of a trampoline, elevators, and moving skywalks. I can assure you, you'll take some time figuring this level out! The game then moves back to the first screen and repeats the sequence. There are a total of five levels for each of the three screens; each of increasing difficulty.

The game is for one or two players and supports joystick use. The disk version comes on a copy-protected, self-booting diskette. One high score is maintained during play, but is not recorded.

A special feature of this game, and one that I have not seen before, is its Modify mode. All of the game's parameters are user-adjustable. That is, you decide how many monkeys you have, how fast the game plays, bonus speed, what score is needed for extra men awards, and if monsters are allowed. This adds a great amount of versatility. If you're just trying to learn the game, you can slow it down, eliminate monsters, and get a feel for the different levels. If you're a real pro, you can speed it up enough to challenge you even after a hundred plays. This is a tremendous feature and it should be added to all arcade games.

Its play is fast, smooth, and error free. Quality Software has produced a game that shows innovation is still possible. By allowing you to pick your own level, from tens of variations instead of one or two, they keep this game alive after many plays.

Mark Renne

Dental Computer Newsletter

E. J. Neiburger, D.D.S., Editor
1000 North Avenue
Waukegan, IL 60085

The D.C.N. is an international group of dentists, physicians and office management people who have interests in office computers. Though the emphasis is on microcomputers, many members use minis. We cater to all makes and brand names.

Annual membership dues $15.00. Membership runs from January to January. If you join mid-year, we will supply you with the year's back issues.

CONVERT-A-DISK

Single-sided disk drive users increase your disk storage by 100%!!!

Convert-A-Disk will work with any 5¼-in. single-sided disk drive no matter what type of computer you are using.

Why pay hundreds of dollars to increase your disk storage?? Convert-A-Disk will allow you to convert all your existing disks to give you a 100% increase disk storage capacity.

Send $15 check or money order to:

CONVERT-A-DISK
P.O. Box 15277
Portland, OR 97214

COMTRONIC SYSTEMS

PMS-1
PROPERTY MANAGEMENT SYSTEM

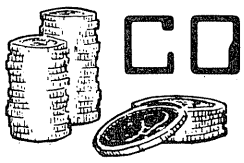
- Saves Hours of Bookkeeping
- Increases Accuracy
- 100% User Support
- User Friendly System
- 350 Units Per Diskette
- Complete Tenant File
- Current Tenant Report
- Late Rent Report
- Late Rent Notices
- Rent Payment History
- Vacancy Report
- Income Statement
- Expense Ledger
- Change of Rent Notices

For TRS-80 Model I and III 32K One Disk Drive.
Excellent for Management Companies or Individuals.
$169.95 on Diskette with Manual, Documentation and Sample Files.

LOAN AMORTIZATION PROGRAM
Prints or displays an amortization schedule for a given mortgage contract. Perfect for Real Estate Investors.

For TRS-80 Model I and III 16K Computers
$12.95 on tape and $15.95 on Diskette

"SATISFACTION GUARANTEED"
COMTRONIC SYSTEMS
4028 Somerset Lane, Kent, WA 98032



COINS

THE UNIQUE PERSONAL COMPUTER PROGRAM FOR COIN COLLECTORS

- **The program that makes cataloging your U.S. coin collection a snap!**
- **Data file with latest Market Value for all grades of most coins supplied on program disk.**
- **Quarterly Market Value updates available.**
- **Multiple printed reports give Inventory Value - Profit & Loss - Tax / Insurance information.**
- **Concise Users Manual included.**
- **Available for 48K TRS-80 I/III 64K IBM-PC.**
- **Write for free brochure or order now.**

VISA* MasterCard

Price - $95 (Calif. residents add 6% Sales Tax)
plus $1.50 to cover postage and shipping

COMPU-QUOTE Telephone 213/348-3662
6914 Berquist Ave. Dept.180 Canoga Park, CA 91307

TRS-80 MODEL III

* BOOKKEEPER *
$34.50
Perfect for small business. TIME DOME has pre-loaded files.

* SCRIPSIT — MAILER *
$34.50
Form letters, envelopes, labels from address list and letter made by SCRIPSIT.

* SERVICE BILLING *
$34.50
Rate chart, serial number, from-to-date.

(714) 774-9383

H.A.K. WORKSHOP
9791 Orange
Anaheim, Cal. 92804

FILEQUEST— Do mailing labels, small inventories. Tape or disk $15.00.

CALENDAR— An electronic appointment book. Printer output. Only for 32K RS disk $10.00.

MONEYPLANNER— See how much you have to save for the future. $7.00.

TORRS INVASION— Board game. 100 square grid. User supplies board and pieces. Earn credits to build your units. Up to 26 pieces. Three levels of play. User can define pieces. Can you beat the computer? $10.00.

DATA MANIA, INC.

51625 Chestnut Road
Granger, IN 46530

Add $1.50 P/H. Specify your system, 16K or 32K and TRS-80 E/B or Model I/III. Write for software list.

FREE

business software directory

Micro Architect, Inc.
96 Dothan St., Arlington,

MA 02174

THE ULTIMATE IN SOFTWARE UNPROTECTION

FPS-3 IS A FRONT PANEL SIMULATOR FOR THE TANDY CORP. TRS80 MODEL III

JUST FLIP A SWITCH AND! ! !

THE PROGRAM IN MEMORY IS COPIED TO YOUR CHOICE OF DISK OR TAPE. TO RUN THE COPY SIMPLY BOOT THE DISK FROM RESET OR LOAD THE TAPE WITH THE SYSTEM COMMAND. YOU DO NOT NEED ANY TECHNICAL KNOWLEDGE TO USE THE FPS-3. ALL YOU NEED TO INSTALL THE FPS-3 ARE A HALF HOUR OF YOUR TIME AND A SCREWDRIVER. THE COST FOR A COMPLETE FPS-3 IS ONLY $50.

WHAT THE SOFTWARE GODS HAVE HIDDEN

THIS LITTLE DEVIL SHALL REVEAL

J.E.S. GRAPHICS, P.O. BOX 2752, TULSA, OK. 74101 CALL 918 742 7104.

ULTRA TERM

The best TRS-80 terminal package at **any** price. Why pay $129 or more?

Split screen, 1K print buffer, half and full duplex support. Direct to disk file transfer permits unattended operation of host computer! Supports RS Modem 2, Hayes Smartmodem, Lynx and more!

Only $59 Postpaid
Send check or money order to:

Computers Unlimited
3174 Hylan Blvd.
Staten Island, NY 10306
(212) 979-6689
MC & Visa Accepted — C.O.D. $3 Extra

Bulletin board

This bulletin board space is available free to individuals with single or unusual items for sale or trade, and for other announcements of interest to the general readership of this magazine. Basic Computing reserves the right to reject any commercial advertising in this section.

These notices are free of charge and will be printed one time only on a space-available basis. Notices will be accepted from individuals or bona fide computer clubs only. All announcements must be typed, contain 75 words or less and include complete name and address.

Will take Mod 4 with two disks and daisy wheel printer as partial payment on Florida homesite. Located in Panama City, FL. Current value $8100. Call Washington state (206) 385-0553.

Model I Software: Utilities, languages, games, etc. Original media and documentation. Save 50 percent or more (all games - $5). Send SASE for list. Also, 80-track MPI-91 single-sided, double-density drive with case and power supply for $285. Stephen Smith, 4396 Rocky River Dr., Apt 12, Cleveland, OH 44135.

Poor Man's Floppy: The JPC TC-8 for the Model I, complete with manuals and UTIL program with TINY monitor. Saves and loads programs to cassette at high speed. Many disk-like features. Perfect upgrade until you can afford disk drives. For only $75. Tom Kilbride, 4117 Lyle Ave., Waco, TX 76710 (817) 754-1803.

Modem for Model I needs no RS-232. Includes 40-pin bus cable, two communications tapes with documentation, all for $50. Bob Green, P.O. Box 419, Clarkston, GA 30021.

Centronics P-1 Printer for Model I/III for $100. Includes manual and Model I cable. JPC TC-8 cassette operating system, with manual, $50. Elcompco 4X expansion bus for the Model I, with one gold plug, $50. Wesley Sarver, 1108 San Miguel Way, Sparks, NV 89431 (702) 358-5202.

Line Printer VIII for only $395. Includes Color Computer switch box and extra ribbons. Dave Edick, 15938 Gramercy Dr., San Leandro, CA 94578.

Exatron Stringy Floppy and starter kit. In good condition, only $100. Harold E. Kautz, Jr., 1115 E. Caracas Ave., Hershey, PA 17033 or call (717) 534-2642 after 6 P.M.

Silicon Valley Color Computer Club meets the first Tuesday of each month at 7:30 P.M. at 5201 Patrick Henry Dr., Sunnyvale, CA. For information call (408) 749-1947 (voice) or (408) 733-6809 (modem).

Color of San Francisco meets the second Sunday of each month at 6:30 P.M. at 141 10th St., San Francisco, CA. Write to P.O. Box 421242, San Francisco, CA 94142-1242 or call (415) 641-1132.

Greenville Color Computer Club meets every Tuesday night 7:30 at the Plain Elementary School in Simpsonville, SC. This newly formed club is already fifty members strong. For information ask for Ed Lowe at (803) 876-3928, 876-3812, or write to P.O. Box 6, Gray Court, SC 29645.

MT32 Expansion Box for the Model I. Contains 32K memory and parallel printer interface. Comes with peripheral adapter allowing another device to be plugged into the keyboard. Only $25. Douglas Stewart, 15 Mountain View Rd., Cape Elizabeth, Me 04107 or call (207) 767-2351 after 6 P.M.

Full System For Sale: Model I, 48K, Percom Doubler, Omikron CP/M board, RS-232, green screen, lowercase, VisiCalc, Scripsit, Microfiles, CP/M, NEWDOS/80. All for $1000. The computer is the latest Model I manufactured by Tandy. Write to B. Stern, Box 69400, Los Angeles, CA 90069 or call (213) 851-7722.

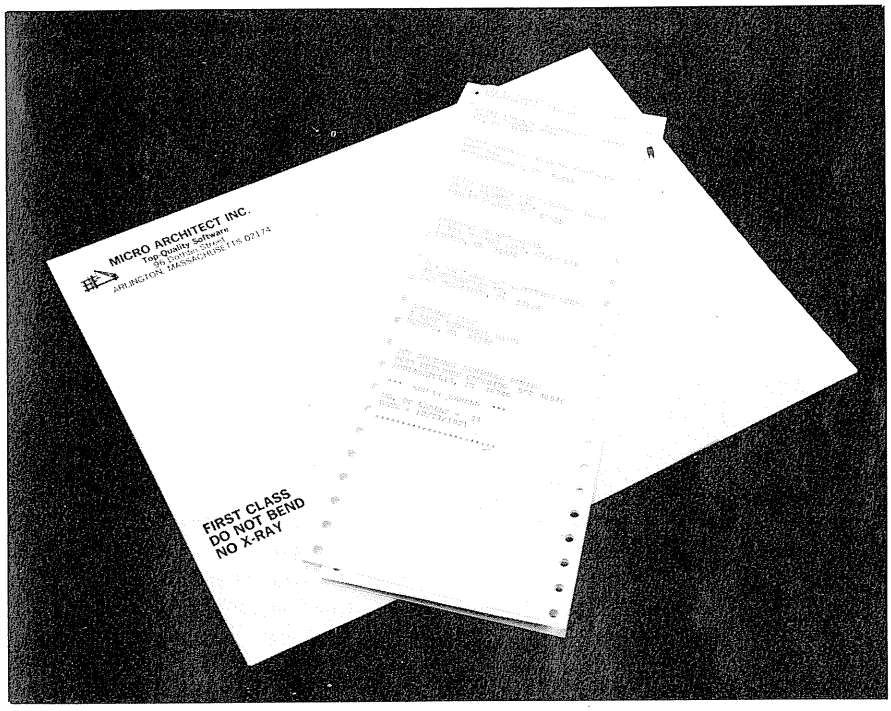
48K Model I with MDX-2 Interface, RS-232, modem, printer port, two Tandon 40-track drives, NEWDOS, FORTRAN and more. $1595. Write to Wendell KR Hutchings, 10787 E. Virginia, Apt D, Aurora, CO 80012 or call (303) 361-6484 for more information.

I want a cassette-based bridge game program for my 48K Model III. Any suggestions where I can find one? Richard Burckhardt, 744 Dorchester, Houston, TX 77022.

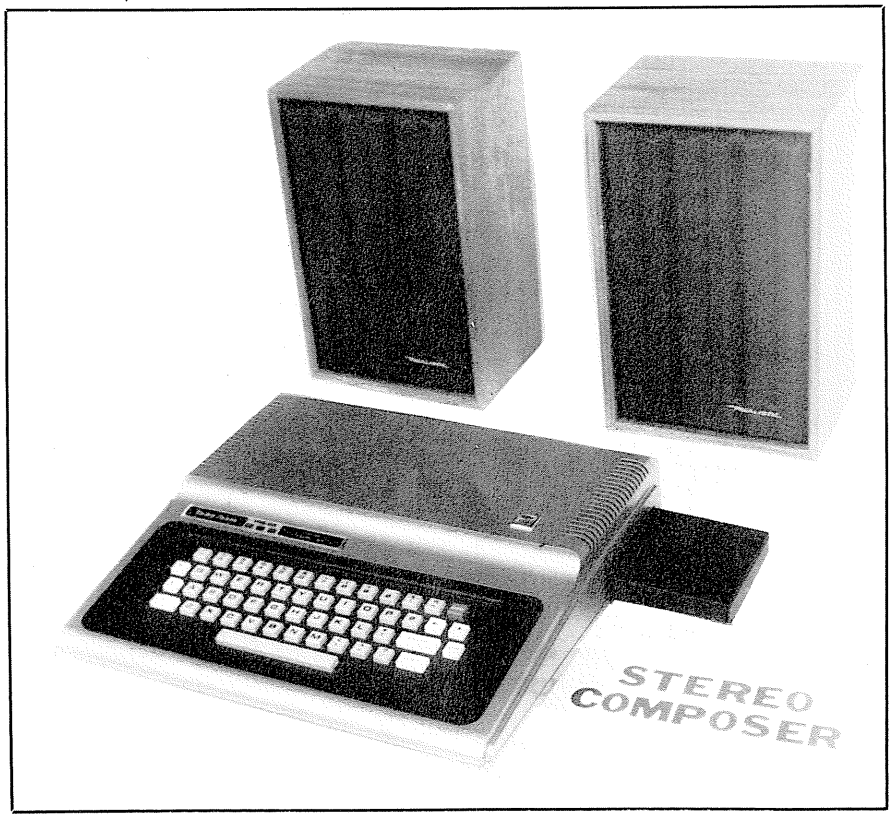
Electronics Equipment: All are non-commercial, used. Denton Model MLA-2500 linear amplifier ($600), RCA Model WP-704B DC power supply ($25), High voltage probe RCA WV-297 ($45), Sine/square wave generator ($100), transistor's curve tracer ($100), and more. Contact Wm. D. Shevtchuk, 1 Lois Ave., Clifton, NJ 07014 or call (201) 471-3798 for more information.

For immediate release

Mail File Converter



Stereo Composer



Mail File Converter

Micro Architect, Inc. now offers a conversion program for users of the Radio Shack Model II Mailing List System. The conversion will allow current users of the Radio Shack system to convert and merge existing files into one Micro Architect Mail-X file. The Mail-X program will allow up to 32,767 records in a single file is available for the Models II, 12, and 16 (Model II mode). The Mail-X program sells for $299 under TRSDOS, CP/M-80 and MS-DOS and the conversion program is $49. Contact Micro Architect, Inc., 96 Dothan St., Arlington, MA 02174 or call (617) 643-4713.

Digital Synthesizer

Computerware has introduced the Synther 7 for the Radio Shack Color Computer or TDP System 100. It will turn the keyboard into a musical instrument and simulates the sounds of many different types of instruments.

The software allows for attack, decay, sustain, and release adjustment. Options allow you to choose the type of sound your computer makes. It is avalable for $21.95 on cassette or $26.95 on diskette, plus $2 shipping and handling. Contact Computerware, P.O. Box 668, 4403 Manchester Ave., Suite 103, Encinitas, CA 92024 or call (619) 436-3512.

Stereo Composer

The Stereo Composer allows the Color Computer owner to program four separate voices with a seven octave range. Two voices are directed to each channel and voices may be moved between channels. Dotted and double dotted notes, as well as eighth, quarter, and standard triplet notes are supported.

The hardware features two 8-bit digital-to-analog converters which drive two audio amplifiers. The output may be connected to your home stereo or external speakers. Two built-in volume controls are

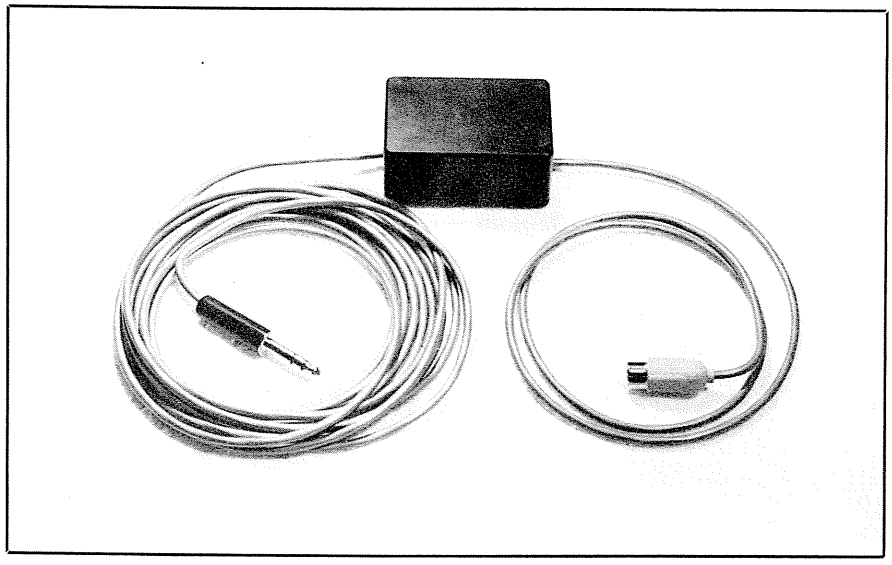
provided. The device does not interfere with the Radio Shack disk controller and will work with any configuration of Color Computer. All hardware and software sells for $119.95. Contact Speech Systems, 38 West 255 Deerpath Road, Batavia, IL 60510 or call (312) 879-6880.

LDOS Utility

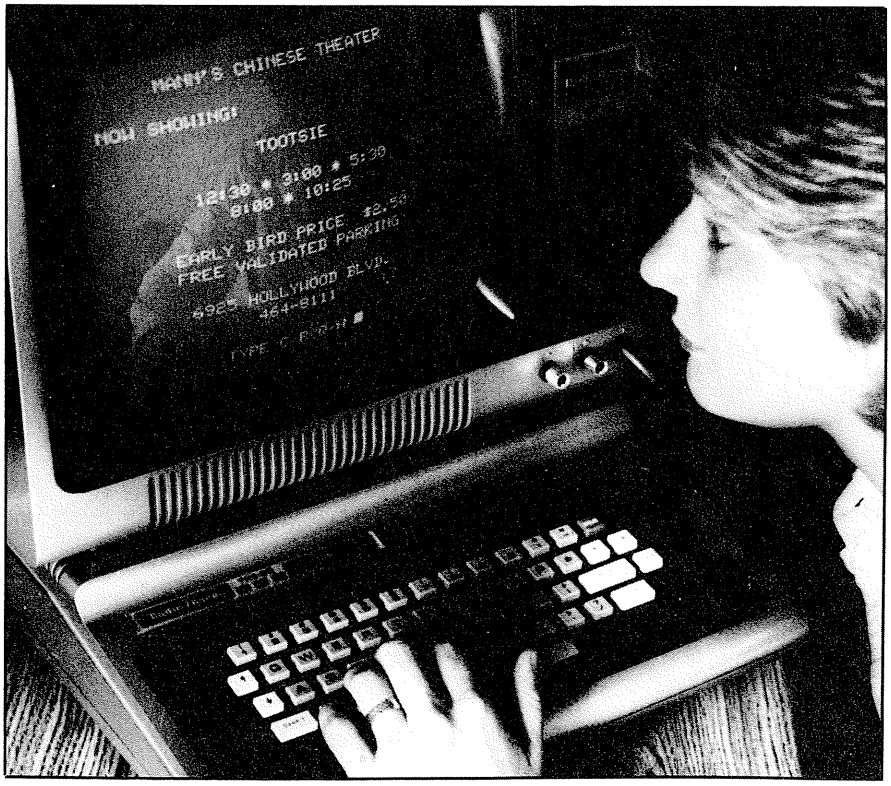
ZSHELL is a command-line preprocessor which adds UNIX-like features to LDOS version 5.1. It supports command-line I/O redirection of standard input (*KI) and standard output (*DO/*PR) devices during a program's execution.

It also allows for the "piping" of standard output from one program to the standard input of another program. The program requires about 1400 bytes of high memory for its use. The package sells for $40 including shipping and handling. Contact MISOSYS, POB 4848, Alexandria, VA 22303.

Kaleidophone



Electronic Yellow Pages



Kaleidophone

The Kaleidophone allows you to connect your Color Computer to any hi-fi and see the music on the TV. It's easy to install and the hardware contains all necessary plugs and cables. Included with the device is a cassette "magazine" of programs for the Kaleidophone. Over a dozen display programs are included. The device with software sells for $49.95 and requires 16K memory, regular or Extended BASIC. Contact Kaleidophone, New Salem Research, West Main Street, New Salem, MA 01355 for more information.

Electronic Yellow Pages

West Los Angeles is now on-line and the phone company isn't involved. BUY-PHONE is a *free* videotext service that contains over 10,000 listings covering all kinds of consumer products, services, and entertainment. The user sees only the information requested and it is arranged by the seller's distance from that consumer.

For example, a person living in Beverly Hills and looking for the movie "Gandhi" would instruct BUY-PHONE to search through its listing for the theaters showing it that day. BUY-PHONE would arrange the information according to distance to that user, display the theater name, show times, prices and other information. Within the data base are over 1,100 indexed restaurants as well as theaters and shops.

BUY-PHONE has begun placing terminals in public locations such as hotel lobbies for use by the general public. It currently covers only West Los Angeles but will be extending its range to other locations.

Orchestra-90

Stereo music and percussion for the Model III are now possible with the Orchestra-90 Special Composer's Edition, a software and hardware product from Software Affair. The Orchestra-90 synthesizes stereo music in four-part

harmony using any combination of trumpet, oboe, clarinet, violin, organ and percussion effects. Five-part harmony is supported on systems with a 4 MHz. clock rate.

A full-screen text editor allows easy transcription of sheet music. No musical knowledge is required. Over 10,000 music files have already been arranged and over 300 are now available on-line from the Orchestra-90 Special Interest Group that is on CompuServe (page HOM-13). The Special Composer's Edition includes sample music, instructions, and the fully-assembled and tested PC board. The high-level stereo output may be connected to any stereo amplifier. The system sells for $99.95 plus shipping and handling. Contact Software Affair, 858 Rubis Dr., Sunnyvale, CA 94087 or call (408) 730-1030.

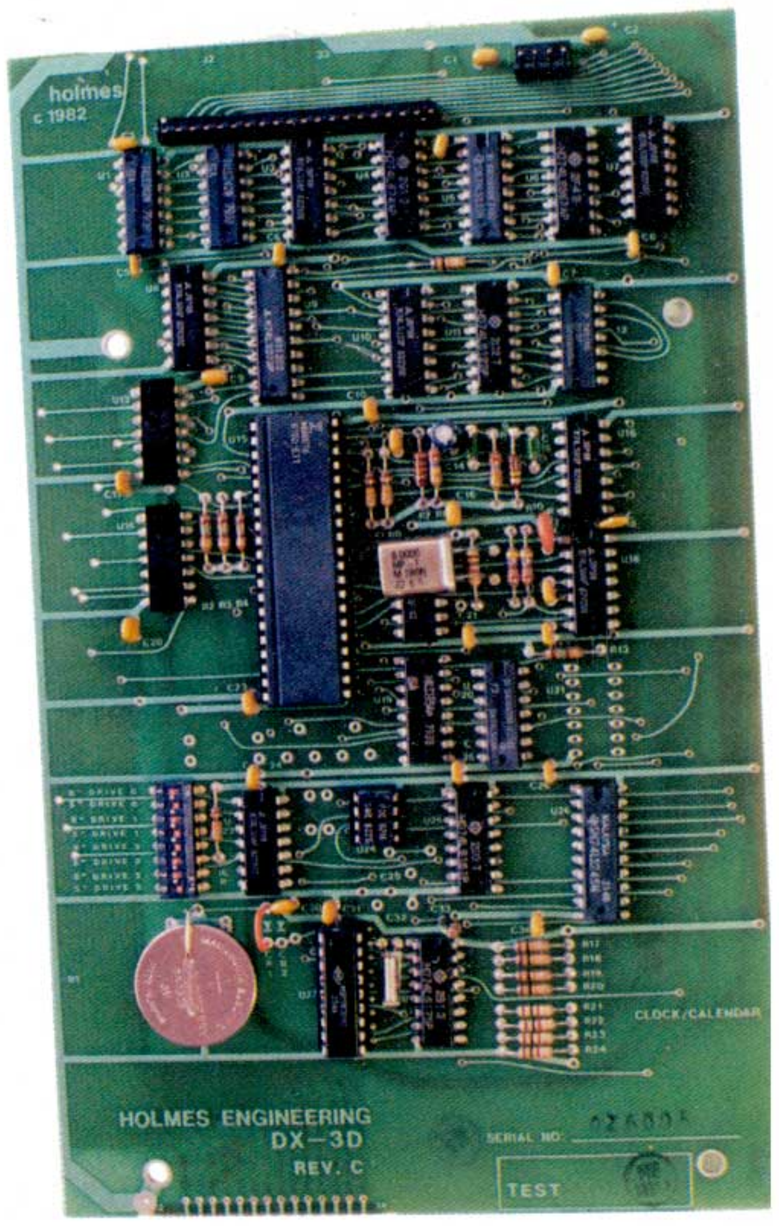
Classical Mosquito!

An original recording of music made by a TRS-80 Model I with the Orchestra-80 synthesizer is available in an Extended Play 45 RPM record. There are eight new, light, and harmonious numbers on this unique album composed by Robb Murray. Records can be ordered for $4 each and orders should be sent to: Robb Murray, 444 St. James Place, Chicago, IL 60614.

Model III Disk Controller Board

The DX-3D is a fully assembled and tested floppy disk controller board from Holmes Engineering. It can be configured to allow the use of 5¼" or 8" drives. For reliable and long-term performance, it uses gold edge-connectors and has fully buffered address and data lines. The DX-3D incorporates advanced digital phase-locked loop circuit design that will not drift or need adjustment. An optional version, the DX-3DC, is available that includes a real-time clock and calendar. For more information, contact Holmes Engineering, Inc., 3555 South 3200 West, Salt Lake City, UT 84119 or call (801) 967-2324.

Model III Disk Controller Board



Advertiser index

Support our advertisers!

AARDVARK L.T.D.

TRS-80 COLOR COMMODORE 64 VIC-20 SINCLAIR/TIMEX TI99/4A

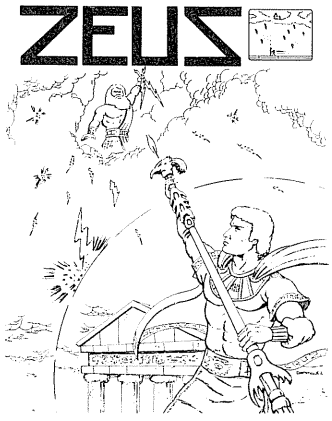
QUEST – A NEW IDEA IN ADVENTURE GAMES! Different from all the others. Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. TRS-80 Color, and Sinclair, 13K VIC-20. Extended BASIC required for TRS-80 Color and TI99/A. $14.95 each.

32K TRS 80 COLOR Version $24.95.

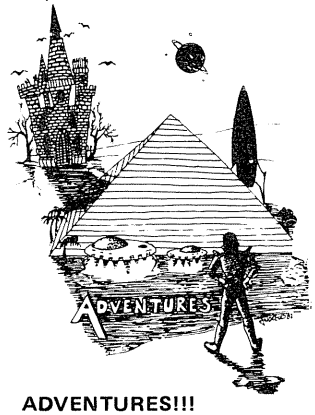
Adds a second level with dungeons and more Questing.

WIZARDS TOWER – This is very similar to Quest (see above). We added wizards, magic, dragons, and dungeons to come up with a Quest with a D&D flavor. It requires 16k extended color BASIC. 13k VIC, Commodore 64, TRS-80 16k Extended BASIC, TI99/A extended BASIC. $14.95 Tape, $19.95 Disk.

Authors – Aardvark pays the highest commissions in the industry and gives programs the widest possible advertising coverage. Send a Self Addressed Stamped Envelope for our Authors Information Package.



ZEUS – It's fast and furious as you become the WIZARD fighting off the Thurderbolts of an angry ZEUS. Your Cone of Cold will destroy a thunderbolt and your shield will protect you – for a while. This is the best and highest speed arcade action we have ever done. Difficulty increases in wave after wave, providing hours of challenging fun and a game that you may never completely master. Commodore 64, Vic20 (16k expander), and 16k TRS-80 Color Computer. (ALL MACHINE CODE!)
$19.95 tape $24.95 disk. (Tape will not transfer to disk.)



ADVENTURES!!!

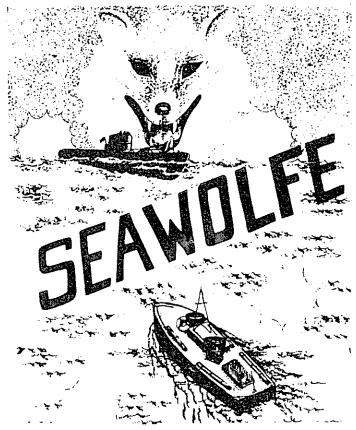
The Adventures below are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are interactive fantasies. It's like reading a book except that you are the main character as you give the computer, commands like "Look in the Coffin" and "Light the torch.")

Adventuring requires 16k on Sinclair, and TRS-80 Color. They require 8k on OSI and 13k on VIC-20. Now available for TI99. Any Commodore 64.

$14.95 Tape – $19.95 Disk.

ESCAPE FROM MARS
(by Rodger Olsen)
This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

PYRAMID (by Rodger Olsen)
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!



SEAWOLFE – ALL MACHINE CODE In this high speed arcade game, you lay out patterns of torpedoes ahead of the attacking PT boats. Requires Joysticks, at least 13k RAM, and fast reflexes. Lots of Color and Sound. A fun game. Tape or Disk for Vic20, Commodore 64, and TRS-80 Color. NOTE: tape will not transfer to disk!

$19.95 Tape – $24.95 Disk.

Dungeons of Death – This is the first D&D type game good enough to qualify at Aardvark. This is serious D&D that allows 1 to 6 players to go on a Dragon Hunting, Monster Killing, Dungeon Exploring Quest. Played on an on-screen map, you get a choice of race and character (Human, Dwarf, Soldier, Wizard, etc.), a chance to grow from game to game, and a 15 page manual. 16k Extended TRS-80 Color, 13k VIC, Commodore 64. At the normal price for an Adventure (14.95 tape, $19.95 disk), this is a giveaway.

Dealers – We have a line of about 100 original programs for the machines listed here. We have High speed Arcades, Quality Adventures, Word processors and Business Software for Small machines. Better yet, we have excellent Dealer support. Phone for information.

Send $1.00 for Complete Catalogue - Please specify system on all orders - $2.00 Shipping Charge on each order

VISA

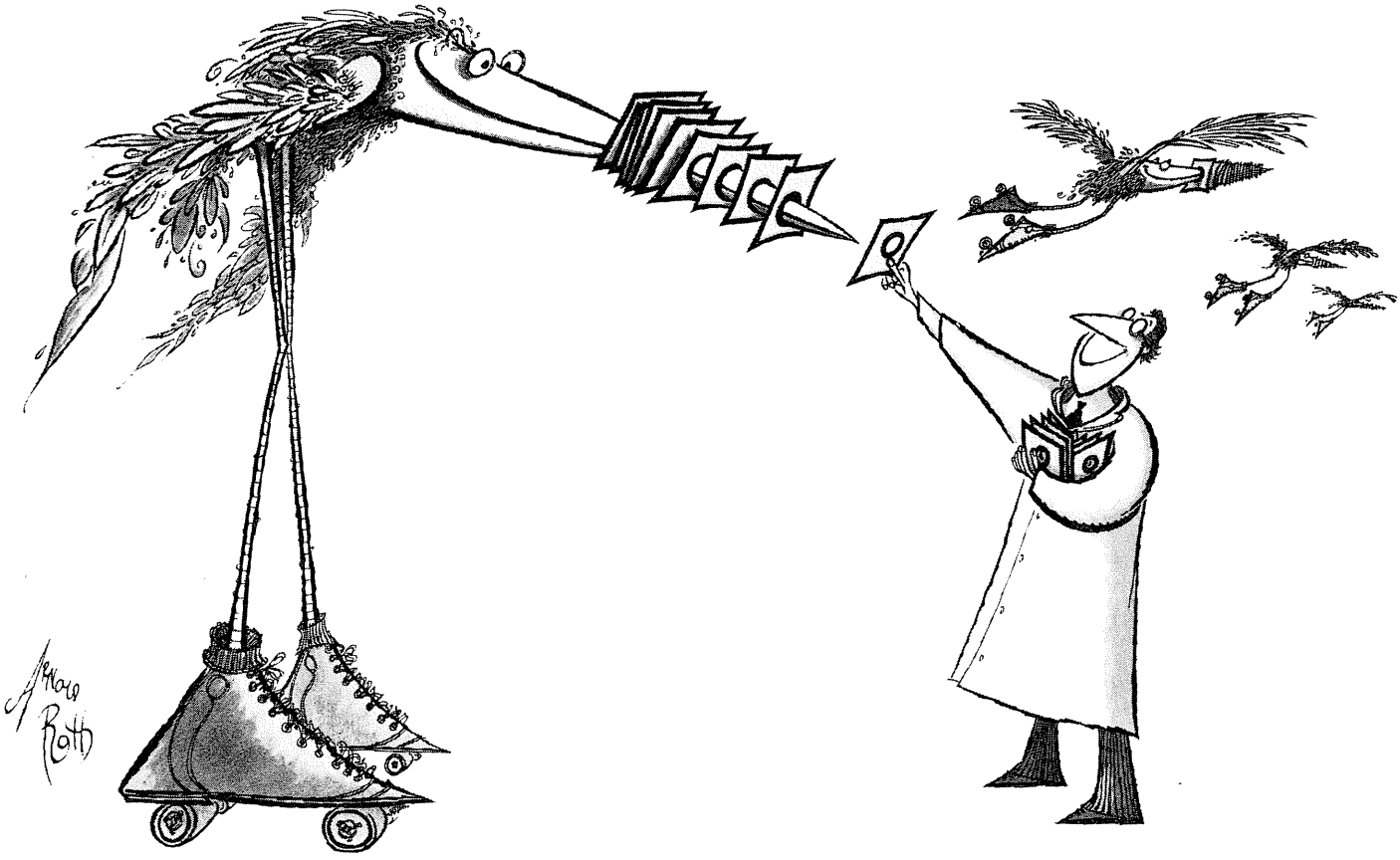
AARDVARK L.T.D.

2352 S. Commerce, Walled Lake, MI 48088 / (313) 669-3110

Phone Orders Accepted 8:00 a.m. to 4:00 p.m. EST. Mon.-Fri.

MasterCard

Smallware™



Our software is making a name for itself.

Smallware. That's what we've named our unique software designed for microcomputers. Smallware offers much more than ordinary software: high quality, customer support and a complete product line. You can buy software anywhere. But for the special features of Smallware, The Small Computer Company is your one and only source.

The Small Computer Company is known to many as the company who developed the filing system software Profile® II, Profile Plus and Profile III Plus for Radio Shack; and filePro,™ our CP/M® version.

Now, whether you're a microcomputer end-user, dealer or manufacturer, you can order our Smallware directly from us.

Here are just some of the enhancements we offer to Model III users:

PROSORT: If you need to select records for a report by more than two criteria (income, zip code, purchases, etc.) Prosort lets you use up to sixteen. Once selected, the records can be sorted by up to five criteria (zip code, within state, by last name). Prosort also offers substantially greater sorting capacity. . . $150

FORMS: If you prepare forms that require several lines of data, from invoices to shipping instructions, Forms is invaluable. It allows you to print individual forms (up to 13″ x 11″) with graphics, trademarks, logos, underlining, subscript and superscript functions. $125

ARCHIVE: Lets you maintain up-to-the-minute, clean files by removing inactive records and transferring them to a pre-determined list or file; split an existing data base into any number of specialized files; free substantial disk storage space $150

PROPACK:™ A tool that lets BASIC programmers more easily customize Profile systems. The resulting programs are shorter, easier to write and faster running. Propack also gives the BASIC program indexed access to Profile data $75

For Model II, 12 and 16 users, there's Quikback™ with Format, Display, Transfer, 8 Line Reports With Math, Math Upgrade for Profile Forms, Math 64, Propack and more.

The Small Computer Company does more than create award-winning Smallware. Our commitment to the customer extends to custom design as well as system consultation.

For further information, call (212) 398-9290. To order, ask for Mr. Sayles.

The Small Computer Company, Inc.

230 West 41st Street, Suite 1200, New York, New York 10036

80 readers' choice micro 1982

An Affordable Alternative

smal-LDOS

VERSION 5.1
THE TRS-80™ OPERATING SYSTEM
MODEL I AND III

First Time Available to the Public Is This Unbelievably Priced Sub-Set* of the Renowned LDOS Operating System.

Order Yours Today for Only **$59.00** plus $3.00 Shipping & Handling

* Note: smal-LDOS does not contain all the features or utilities of the original "FULL LDOS" system.

* Model-I smal-LDOS provided on 35 track single density media.
* Model-III smal-LDOS provided on 40 track double density media.
* smal-LDOS can be provided on special media configurations at an additional charge.
* Prices & Specifications are subject to change without notice.
† Although not required, LSI recommends two or more drives when using smal-LDOS.

DEALER INQUIRIES WELCOME. smal-LDOS is a product of LSI. TRS-80 is a trademark of Tandy Corp.

LOGICAL SYSTEMS INC.

Logical Systems, Inc. 8970 N. 55th P.O. Box 23956 Milwaukee, WI 53223 (414) 355-5454

All You Want in a Computer At a Price You'd Never Expect

Introducing the New TRS-80® Model 4 Personal Computer From Radio Shack

64K Model 4
With Two Disk Drives
And RS-232C

1999.00

Cat. No. 26-1069

- High-Performance 4 MHz Z80A
- Runs Thousands of Model III TRSDOS®, LDOS™ and CP/M® Plus Applications
- 80-Column by 24-Line Video Display With Upper and Lower Case
- 64K RAM Is Expandable to 128K—Use it As a Super-Fast RAM-Based "Disk Drive"
- Advanced Job Control Language Allows Completely Unattended Operation
- Sound is Accessible From BASIC or JCL
- Includes Print Spooler (Disk or Memory)
- 70-Key Keyboard Includes Datapad, Plus CONTROL, CAPS and 3 Function Keys
- New Microsoft BASIC Includes CHAIN And COMMON for Larger Programs
- Communications Package Allows Model 4 to be Used As Host or Terminal
- Forms Control, Routing and Linking, and Filtering Using Advanced TRSDOS 6.0
- Reverse Video Highlights Corrections
- Options Include 640 x 240 Graphics, Hard Disk System and CP/M-Plus*

With its new advanced operating system and Microsoft® BASIC, Model 4 is already an amazing value. But consider these other built-in features: two 184K, 5¼" double-density disk drives, hi-res monitor, parallel printer and RS-232C interfaces. Don't pay more later to add these features to another personal computer. We *include* what most charge extra for!

See the fantastic new TRS-80 Model 4 at your nearby Radio Shack store, Computer Center or participating dealer—today.

Radio Shack®
The biggest name in little computers®
A DIVISION OF TANDY CORPORATION

Send me a free TRS-80 Computer Catalog today!
Mail To: Radio Shack, Dept. 84-A-15
300 One Tandy Center, Fort Worth, Texas 76102

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

*Available soon! Retail prices may vary at individual stores and dealers. LDOS is a trademark of Logical Systems, Inc. CP/M-Plus is a registered trademark of Digital Research, Inc. Microsoft is a registered trademark of Microsoft, Inc.