Hardware Project Issue

A CWC/I PUBLICATION
JULY 1984 USA $2.95 CAN $3.50

HOT CoCo

THE MAGAZINE FOR TRS-80 COLOR COMPUTER® AND MC-10® USERS.

Build Your Own Ultimate Interface

J&M's Disk System: The Right Stuff?

Do-It-Yourself Lowercase Mod

How Great Is Graphicom? The Pictures Tell You

Plus: PBJ's Word-Pak— Get a Screen Like the Big Boys'

0 07
A 74470 12067

From Sesame Street to Outer Space...

Radio Shack Has and Entertainment

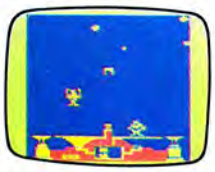
The CTW Software Group, a division of Children's Television Workshop, brings you ten game-style educational programs. Each one encourages children to experiment, explore and solve problems while having fun.

For TRS-80® Color Computers with Extended BASIC

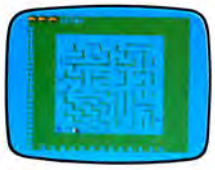
19.95 Each

Taxi.* Kids earn fares and tips as they drive through six cities from New York to Shanghai. A Cooperative Strategy Game for ages 7 and up. #26-2509.

Peanut Butter Panic.* The sky's the limit as players cooperate to catch stars, make sandwiches, and win. A Cooperative Strategy Game for ages 7 and up. #26-2523.



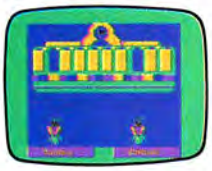
Star Trap.* Players must race through a maze to trap a slippery star before time runs out! A Cooperative Strategy Game for ages 7 and up. #26-2510.



Cookie Monster's Letter Crunch.™** It's Cookie Time! Help Cookie Monster match words and letters to bake and eat cookies! A Basic Skills Game for ages 3-6. #26-2526.

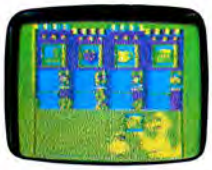
*Joysticks required. **Cassette recorder required.
***Joysticks and cassette recorder required.

Grover's Number Rover.™** Grover's rover is ready to blast off! Hop aboard and help him play with Twiddlebugs and numbers! A Basic Skills Game for ages 3-6. #26-2522.



Ernie's Magic Shapes.™** Ernie wears the top hat, but you're the magician. Help Ernie match shapes and colors in six different ways. A Basic Skills Game for ages 3-6. #26-2524.

Big Bird's Special Delivery.™** Help Big Bird deliver the mail! Match the pictures and bring each package to the right store. A Basic Skills Game for ages 3-6. #26-2525.



Grobot.* How well will your astro-garden grow? Plant, protect and harvest—it's up to you and Grobot. A Creative Exploration Game for ages 10 and up. #26-2527.

Time Bound.* Race through time and learn about history, in hot pursuit of your hapless assistant, Anacron. Creative Exploration Game for ages 10 and up. #26-2528.

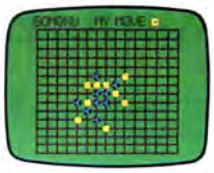
Flip Side.* Stake your claim, surround the squares, and watch the screen flip colors! Planning is the key. A Creative Exploration Game for ages 10 and up. #26-2529.

the Educational Software You Want.

Why feed quarters into video game machines when you can bring arcade-style thrills into your own living room with Radio Shack's exciting Color Computer games. They can provide hours of fun for the whole family.

Low As 19.95

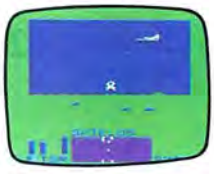
Gomoku and Renju. The classic oriental game of strategy! Block your opponent while attempting to place five of your own men in a row Hours of fun. #26-3069. **$19.95**



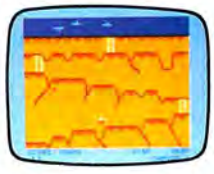
Slay the Nerius.* Defend your submarines against deadly starfish and the ancient seaworm—the fearsome Nerius, a creepy nemesis from the Deep. #26-3086. **$24.95**

Double Back.* As you "double back" to catch your own tail, try to encircle the "safe" screen objects to gain points in this tricky game. Challenges mount as you play. #26-3091. **$19.95**

Star Blaze.* Protect the Milky Way! Radar shows menacing vessels nearby. Seek, destroy and check radar again. Red alert! There's no let up in the excitement. #26-3094. **$19.95**



Canyon Climber.* An action game with a difference. As a cliff hanger, you're challenged by one test after another—kicking goats, zinging arrows and falling objects! #26-3089. **$34.95**



Dungeons of Daggorath.* You're pitted against a succession of awesome beasts. Each victory brings you closer to your ultimate opponent—the evil wizard! #26-3093. **$29.95**

Baseball. Nine innings of fun! You're in full control of this realistic simulation of America's Number One sport, both behind the plate and on the field. #26-3095. **$24.95**

ZAXXON.* The official home version of the great arcade favorite by Sega! Match wits with the deadly ZAXXON Robot! Challenges escalate as you progress. 32K required. #26-3062. **$34.95**

Available at over 1100 Radio Shack Computer Centers and at participating Radio Shack stores and dealers

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

NEW 1984 TRS-80 CATALOG RSC-11.
Send me a free copy today.

Mail To: Radio Shack
Dept. 84-A-925
300 One Tandy Center
Fort Worth, Texas 76102

NAME _____

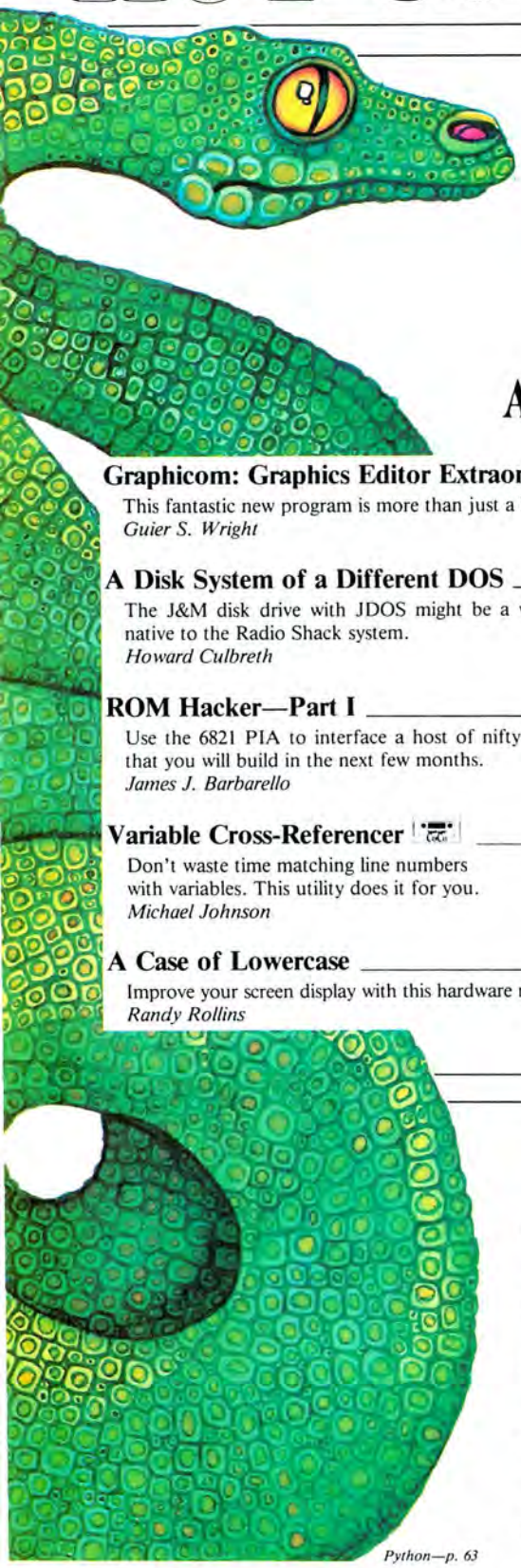
ADDRESS _____

CITY _____ STATE ____ ZIP _____

TELEPHONE _____

Prices apply at participating Radio Shack stores and dealers. Muppet characters are trademarks of Muppets, Inc. All rights reserved. ZAXXON is a registered trademark of Sega licensed to Datasoft, Inc.

HOT CoCo



ARTICLES

This symbol indicates the program's placement on the Instant CoCo loader, available on cassette. See our Instant CoCo ad for details.

Cover art by Wayne McLaughlin

DEPARTMENTS

Python—p. 63

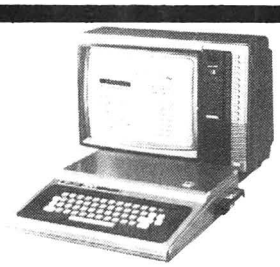
Vol. 2 No. 2 July 1984

HOT CoCo (ISSN 0740-3186) is published monthly by Wayne Green Inc., 80 Pine St., Peterborough, NH, and additional mailing offices. Subscription rates in U.S. are $24.97 for one year, $38 for two years, and $53 for three years. In Canada and Mexico, $27.97—one year only, U.S. funds. Application to mail at second class postage rates is pending at Peterborough, NH and additional mailing offices. Nationally distributed by International Circulation Distributors. Foreign subscriptions (surface mail), $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail), please inquire. In South Africa contact *HOT CoCo*, P.O. Box 782815, Sandton, South Africa 2146. **All subscription correspondence should be addressed to *HOT CoCo*, Subscription Department, P.O. Box 975, Farmingdale, NY 11737. Please include your address label with any correspondence.** Postmaster: Send address changes to *HOT CoCo*, Subscription Services, P.O. Box 975, Farmingdale, NY 11737.

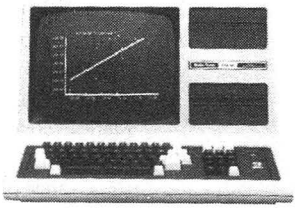
From Computer Plus to YOU . . .

PLUS *after* PLUS *after* PLUS

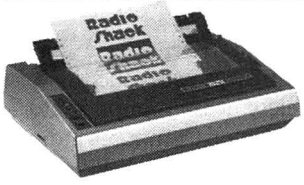
Model 100 8K $679
Model 100 24K $835



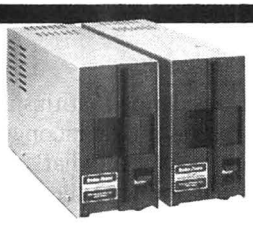
Color Computer II 16K $135
w/16K Ext. Basic $165
w/64K Ext. Basic $210



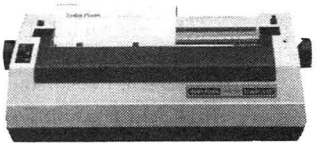
Model 4 16K $849
Model 4 64K
2 Disk & RS232 $1699



DMP120 $395
DMP200 $520



Color Computer Disk Drive
Drive 0 $329 Drive 1 $235



DWP210 $629

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Model 4 Portable 64K w/2 Drives	1525
Model 2000 2Dr	2299
Model 12 1 Drive	2360
Model 16B 1Dr 256K	3965
MODEMS	
Hayes Smartmodem II	225
AC-3	129
DC Modem I	89
DC Modem II	160
PRINTERS	
Silver Reed EXP500 D.W. Ser.	455
Silver Reed EXP550 D.W. Par.	525
CGP115	159
CGP220 Ink Jet	545
DMP110	305
DMP420	735
Toshiba 1340 (24 wire head)	779
Gemini 10X	289
Gemini 15X	409
CITOH Prowriter	359
Okidata	CALL
Epson	CALL

ETC.	
Disk Drive Controller	139
Extended Basic Kit	39.95
PBH Ser/Par Conv.	69
64K Ram Chips	62.95
Deluxe Keyboard	35.95
Superpro Keyboard	69.95
HJL Keyboard	79.95
CCR-81 Recorder	52
Deluxe Joystick (each)	35.95
Joysticks (pair)	22
Video Plus (monitor adapter)	24.95
Video Plus IIC	39.95
Amdek Color 1+ Monitor	299
BMC Color Monitor	255
BMC Green Monochrome Monitor	99
Taxan Green Mono. Monitor	130
Taxam Amber Mono. Monitor	139
SOFTWARE	(Tape Version)
Zaxxon	34.95
The King	26.95
Trap Fall	27.95
Buzzard Bait	27.95
Devil Assault	27.95

Colorpede	29.95
Juniors Revenge	28.95
Pac Attack	24.95
Block Head	26.95
Froggie	24.95
Lunar Rover Patrol	24.95
Lancer	24.95
Typing Tutor	23.95
Galagon	24.95
Scott Adams Adventures	19.95
Sea Dragon	34.95
Colorcome	49.95
Telewriter 64	49.95
O-Pak (disk)	34.95
Key-264K	39.95
Elite-Calc	59.95
VIP Writer	59.95
VIP Calc	59.95
VIP Terminal	49.95
VIP Database (disk)	59.95

Order any 2 software pieces listed and take 10% off their listed price. All Radio Shack software 10% off list. Send for complete list.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA AMERICAN EXPRESS MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

18

TRS-80 is a registered trademark of Tandy Corp.

HOT CoCo

EDITOR-IN-CHIEF
Michael E. Nadeau
MANAGING EDITOR
Janet Fiderio
REVIEW EDITOR
Mark E. Reynolds
NEW PRODUCTS EDITOR
Cynthia Smith
TECHNICAL EDITORS
Peter Paplaskas,
Guier Wright,
Keith Johnson (Instant CoCo)
EDITORIAL DESIGN MANAGER
Susan Gross
EDITORIAL DESIGNERS
Susan Hays, Maurelle Godoy
PROOFREADERS
Director: Peter Bjornsen
Harold Bjornsen, Robin Florence
RESEARCH ASSISTANT
Celeste Wrenn

PUBLISHER/EDITOR
Wayne Green
VICE PRESIDENT/
GENERAL MANAGER
Debra Wetherbee
VICE PRESIDENT/FINANCE
Roger Murphy
EDITORIAL DIRECTOR
Jeffrey DeTray
DIRECTOR OF MARKETING
AND SALES
David Schissler
CIRCULATION DIRECTOR
William P. Howard
603-924-9471
ASSISTANT CIRCULATION
MANAGER
Frank S. Smith
BULK & NEWSSTAND
SALES MANAGER
Ginnie Boudrieau
1-800-343-0728
ADVERTISING, 603-924-7138
Director: Stephen Twombly
Sales Manager: Raino E. Wirein,
Sales Representative: Barbara Alvarez
Ad Coordinator: Suzanne DesRochers
CW Communications/Peterborough
West Coast Office
160 Marsh Road
Menlo Park, CA 94025
415-328-3470 or 3471
Sales Manager: Giorgio Saluti
Sales Representatives: Alison Walsh,
Karen Letendre
PUBLIC RELATIONS
Jim Leonard

DIGRESSIONS

Where Have All The Hardware Hackers Gone?

Considering that this is our hardware project issue, the title of this piece seems a little incongruous. Nonetheless, it is an important question to address in light of what's happening in the Color Computer industry.

I really don't see the number of hardware hackers dwindling. Rather, I see them being increasingly outnumbered by new Color Computer owners—those who have purchased their machines in the last six to eight months. These new owners are mostly first-time computer users whose interests lie far from delving into their Color Computers with a soldering iron.

This is understandable; after all, everyone does crawl before they walk. But how many will learn how to "walk"?

Not many. Some just want to run a few applications, some want to enhance their children's education, some want to be entertained, and some don't know what they want. Few will learn more than the simplest programming skills; fewer still will learn the hardware end.

What this means to aficionados of hardware hacking is that the available literature will become less technically oriented, and that they will be more and more on their own to come up with ideas for new projects.

But the hackers among you can do something about this. There is a dearth of good hardware-oriented literature targeted at the rank beginner. We need articles that literally tell you which end of the soldering iron gets hot. Remember, the simplest instructions for an experienced hardware hacker can intimidate the uninitiated.

We hope that this group of new users yields a good crop of hardware experts. But we need a little enticement from those of you who are already experts. Think back to the days when you were just starting out and share some of the advice that you found helpful. It's the least you can do for your hobby.

Changes, Changes

We've made some changes in regards to our columns. First, we've discontinued two: Elmer's Arcade and Graphically Speaking. Those of you who had been following these columns, take heart. We intend to make use of both authors' talents in other ways in future issues.

Re:FLEX, soon to be renamed The DOSsier, is now running monthly—good news for you users of OS-9, Flex, and Star-DOS operating systems. Also, we've reintroduced The Basic Beat, which was a big favorite of beginners of all ages.—*M.N.* ■

PRODUCTION
Director: Nancy Salmon;
Lahri Bond, Cindy Boucher,
Linda Drew, Donna Hartwell,
Laurie Jennison, Star Kachadoorian,
Marlene Mowbray, Kenneth Sutcliffe,
Leslie Walden, Theresa Verville,
Robert M. Villeneuve, Lynne Simonson
Ad Coordinators: Patricia Bradley, Paula
Ramsey; Assistant: Jean Southworth
Advertising Production: Fiona Davies,
Bruce Hedin, Michael Ford, Jane Preston
CHIEF COPYWRITER
Steve Tripp
HOT CoCo, Louis Marini
PHOTOGRAPHY
Supervisor: Nathaniel Haynes;
Sandra Dukette, Laurie Gardos,
Jeanne Quickmire, Sturdy Thomas
TYPESETTING
Supervisor: Dennis Christensen;
Darlene Bailey, Marie Barker, Prem
Gongaju, Lynn Haines, Cynthia
Letourneau, Kimberly Nadeau,
Debbie Nutting, Lindy Palmisano,
Heidi N. Thomas
DESIGN
Manager: Joyce Pillarella;
Design Consultant: Dion Owens
CREATIVE DIRECTOR
Christine Destrempes

The left bracket, [, replaces the up arrow used by Radio Shack to indicate exponentiation on our printouts. When entering programs published in *HOT CoCo*, you should make this change.

Article submissions from our readers are welcomed and encouraged. Inquiries should be addressed to: *HOT CoCo* Submissions Editor, 80 Pine Street, Peterborough, NH 03458. Include an SASE for a copy of our writer's guidelines. Payment for accepted articles is made at a rate of approximately $50 per printed page; all rights are purchased. Authors of reviews should contact the *HOT CoCo* Review Editor, 80 Pine Street, Peterborough, NH 03458.

Subscriptions:

Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *HOT CoCo*, Subscription Department, P.O. Box 975, Farmingdale, NY 11737.

Change of Address: Send old label or copy of old address and new address to: *HOT CoCo*, P.O. Box 975, Farmingdale, NY 11737. Please give eight weeks advance notice.

Dealers: Contact Ginnie Boudrieau, Bulk Sales Manager, *HOT CoCo*, Pine St., Peterborough, NH 03458. (800) 343-0728.

Problems with Advertisers: Send a description of the problem and your current address to: Magazine, Rt. 101 & Elm Street, Peterborough, NH 03458. ATTN.: Rita B. Rivard, Customer Service Manager. If urgent, call 1-800-441-4403.

HOT CoCo is a member of the CW Communications/Inc. group, the world's largest publisher of computer-related information. The group publishes 44 computer publications in 18 major countries. Nine million people read one or more of the group's publications each month. Members of the publication group include: Australia: *Australasian Computerworld, Micro Magazine*; Argentina: *Computerworld/Argentina*; Brazil: *DataNews, MicroMundo*; Denmark: *Computerworld/Danmark, MikroData*; France: *Le Monde Informatique*; Germany: *ComputerWoche, MicroComputerWelt, PC Welt*; Italy: *Computerworld Italia*; Japan: *Computerworld Japan, PC Japan*; Mexico: *Computerworld/Mexico*; Norway: *Computerworld Norge, MikroData*; People's Republic of China: *China Computerworld*; Saudi Arabia: *Saudi Computerworld*; Spain: *Computerworld/Espana, MicroSistemas*; Sweden: *ComputerSweden, MikroDatorn, Min Hemdator*; United Kingdom: *Computer Management, Computer Business Europe*; United States: *Computerworld, HOT CoCo, inCider, InfoWorld, Micro MarketWorld, Microcomputing, PC World, 80 Micro, RUN,* and *jr.*

Instant CoCo

Instant CoCo Directory—July

SIDE A

ARTICLE NAME/AUTHOR	FILE	PAGE #	SYSTEM
Copyright Statement	TITLE	---	All
Variable Cross-Reference/Johnson	CROSSREF	30	16K Ext
Anatomy of an Assembly-Language Game—Part II/Meehan (m)	CROAKER2	52	32K Ext
Python/Siddely	PYTHON	63	16K Ext
Color Blockout/Knudson	COLBLOCK	66	4K

SIDE B

A Collectors Item (Sort Of)/Reeves	HATSORT	70	16K
Turbo Stick/Sachetti	TURBO	72	16K Ext
Arcade Action Without Machine Language/Wick	SAVELOND	80	16K Ext

The symbol (m) in the Article Name column indicates the program is machine-language and must be loaded using the CLOADM command. Additional preparatory commands are listed under the article name where appropriate. CSAVEM addresses are listed for your use with the machine-language programs.

If Our Programs Don't Work

Having trouble entering our listings from the magazine? Here are a few tips that might help.

First, we print all our Basic listings in the CoCo's 32-column format. This means that each line should appear the same on the screen as it does in the magazine. If a line on your screen does not match the same line in the magazine, reread what you typed; you might have made an error.

Second, make sure the program is for your computer. Read the System Requirements box. The information in this box represents the minimum system configuration needed to run that particular program. Also, read the article thoroughly before typing in the program. Sometimes the article contains instructions vital to making the typed-in listing work. For instance, some CoCos will not accept the high-speed POKE (POKE 65495,0). The article for a program using this POKE will tell you to change those POKEs to 65494,0 if your computer will not work at the faster speed.

Some CoCos are sensitive to spacing in the program lines. Occasionally a computer will read a line such as FORR = 1TO20 incorrectly, interpreting the FOR not as a keyword, but as a variable. If you've removed spaces from a program listing to save space, and that program will not work, reinsert those spaces.

If everything is okay so far, check the published listing with what you've typed. Common typing errors include confusing a zero with the letter O, a one with the letter I, or a colon with a semicolon. DATA statements are particularly tricky because of the long lists of numbers. Be very careful with these.

Anyone who owns the new CoCos with the 1.2 ROMs, have noticed poor keyboard response in some published programs. To solve this, you can insert this line: FOR Z = 1TO4:POKE340 + Z,255:NEXT after any line that makes reference to PEEK 338-345.

This loop will slow down a Basic program. Another way is to directly insert a POKE xxx,255, where xxx is any keyboard location between 338 and 345. Example: IF PEEK(341) = 251 THEN Y = Y – 1. Change to: IF PEEK(341) = 251 THEN POKE341,255:Y = Y – 1.

Assembly listings usually require an editor/assembler to enter them into your CoCo. The two most common editor/assemblers are Radio Shack's EDTASM+ and The Micro Works' SDS80C. An Assembly listing assembled using the SDS80C will probably not run under EDTASM+.

If all the above fails, send us a printout or a detailed description of the problem you experience along with any error messages. We'll try to work it out for you. We cannot help you if you have modified the original program in any way. ■

Feedback

Give Both Your Disk Drives One-Track Minds

Dennis Elfert's article, "Give Your Disk Drive a One-Track Mind" (*HOT CoCo*, March 1984, p. 54), offered a most interesting idea for saving blown disks, but I was a little disappointed that I couldn't use the program with my two-drive system.

To use the program with two drives, delete lines 90, 95, 135, and 140 and eliminate the prompts. Then make the following line changes:

```
195 DSKO$ 1,T,X,D$(Y),E$(Y):Y=Y+1:
  NEXTX:GOTO55
340 DSKI$ 1,T,X,D$(Y),E$(Y)
540 DSKI$ 1,V,X,D$,E$
1000 DSKI$ 1,0,1,A$(1),A$(2):DSKI$ 1,0,2,
  A$(3),A$(4)
2000 DSKO$ 1,0,1,A$(1),A$(2):DSKO$ 1,0,2,
  A$(3),A$(4)
```

Then insert the following line:

```
53 INPUT"PUT YOUR OBJECT DISK IN
DRIVE 0 AND YOUR D. D. INTO DRIVE 1
AND HIT ENTER";Q$
```

Now you're ready to run the program with two drives.

Keith Langill
West Trenton, NJ

Time Out

There's an error in my article, "It's Sidereal Time" (*HOT CoCo*, May 1984, p. 100). Line 70 should be 70 YY = 84.

Al Burzynski
San Antonio, TX

Reader's Forum Slips

Eds. note—We've made two slipups in the May 1984 Reader's Forum section. In Charles Werner's "Tape EDTASM+ To Disk" (p. 129), change the CSAVEM "EDTASM+", &H3DFF, &H1600 to CSAVEM "EDTASM+", &H1600,&H3DFF,&H1600.

And here is Table 1 of Andrew Halter's "POKE, PEEK and ASCII" tip, for those of you who are still looking:

X = PEEK,POKE value
X$ = ASCII value
If X is greater than or equal to zero, and X is less than 32, then X$ equals CHR$ (X + 96)
If X is greater than 95, and X is less than 128, then X$ equals CHR$ (X – 64)

Table 1. Formula to Find Correct POKEs for ASCII and Graphics Characters.

Color Basic Compiler Review Revisited

On page 38 of the April 1984 *HOT CoCo*, you published a review of Computerware's Color Basic Compiler. Though the review was well written and informative, there was one error I would like to correct.

Mr. Parker claimed that "there is no PEEK instruction." But the Color Basic Compiler does support the PEEK function, as page 5 of the manual shows.

In addition, we are proud to announce that we've expanded the Color Basic Compiler to include string handling. The new commands are as follows: MID$, LEN, CHR$, STR$, INKEY$, ASC$. We've also enhanced these commands to work with strings: DATA, READ, DIM, IF/THEN, PRINT, and PRINT@. The price is still $39.95.

Thanks for passing along this important information.

Sue Searby
Computerware

Single-Sheet Scripsit

I have a 64K Color Computer 2, one disk drive, a DWP 210, and a Color Scripsit disk. I'd like to manually insert a new sheet of paper into the printer after it has printed a page, but Scripsit won't let me. I have to use fanfold paper whenever the document is longer than one page.

The people at Fort Worth tell me that there isn't any patch to solve my problem, and they suggest that I save each page as a separate document. That isn't very satisfactory.

Does anyone have a better solution?

Monica Beukenkamp
P.O. Box 2191
George Town
Grand Cayman
British West Indies

Rating Auto Insurance?

I'm a new 16K Color Basic CoCo 2 owner. I'm also a producer in a non-automated insurance agency and could use a program to rate auto insurance on the CoCo, and perhaps on Radio Shack's Pocket Computer. I'd like to hear from anyone who has such a program.

I enjoy your fine magazine. It's been a great help to this computer rookie.

Mark Ross
3 Margaret Place
Batavia, NY 14020

Music Processor

I own a CoCo with 1.1 ROM, a single disk drive, and a BMC (Epsoncopy) printer (Model BX-80). I am looking for a program that will give me a professional-quality printout of a musical score. It doesn't necessarily have to have any playback or sound capabilities—in essence, what I need is a "note-processor."

Does anyone know of such a program?

David Makower
8 Barbara Road
New City, NY 10956

Swiss CoCo Responses

Eds. note—In the April 1984 Feedback, we published a letter from Andreas Luecke in which he asked if he could use a Color Computer with a voltage adapter and an American TV

Send your letters to Feedback, HOT CoCo, *80 Pine St., Peterborough, NH 03458.*

in Switzerland. Several readers sent in detailed and helpful replies.

Irish CoCo

I assume Mr. Luecke is concerned by the fact that most of Europe operates on 220–240V/50Hz, but I've been using my CoCo in the Republic of Ireland with a 240–110V stepdown transformer and a U.S. TV for three years with no problems, under the most diabolical conditions—voltage supplies down to 190V and strong spikes and surges. The CoCo is a very resilient machine.

Most concerns when operating U.S. equipment in Europe center around the difference in line frequency: 50 cycles in Europe vs. 60 in the U.S. This seems to mean that U.S. motors run slightly slower in Europe, and nonmotor equipment runs slightly hotter at the same indicated voltage as 110. I'm told this is because the indicated voltage is R.M.S. (Root Mean Squared)—not the average.

So, check your transformer-output voltage. 100–150V is optimum. Do not use the small electronic plug-in voltage conversions—they don't work. Take along a line filter. It's good security against spikes and surges. If you use tape or disk for storage, the machines might run a bit slower, and this might affect loading data that was formatted at 60Hz, but I haven't had any

Clubhouse

Have a Color Computer club? Reach prospective members through a note to Feedback.

Western Mass.

The 6809ers is a Color Computer club that meets once a month in the western Massachusetts area. Contact me for details.

Paris Nepus
93 Grochmal Ave. Lot #90
Springfield, MA 01151

Greater Boston

The Greater Boston Super Color User's Group meets the second Thursday of each month at Sylvania Technical School, 63 Second Ave., Waltham, MA. We publish a monthly newsletter, "The Scugbug." Contact me for more information.

John DeBay
100 Central St.
Waltham, MA 02154

Harrisburg, PA

CAPATUG (Capital Area TRS-80 User's Group) meets the first Thursday of each month at the Fairview Township Fire House in New Cumberland, PA. For more information, phone the CAPATUG BBS at 717-774-6543, or write CAPATUG, 340 Lewisberry Road, New Cumberland, PA 17070.

David Morrow
Vice President

Jefferson City, MO

We are forming a Color Computer User's Group in the Jefferson City/mid-Missouri area. Those interested should contact me for information.

CoCoMUG
c/o Wayne Johnson
900 Rock Hill Road
Jefferson City, MO 65101
314-893-2789

London, Ontario

The London CoConuts Computer Club meets at 7 p.m. on the last Monday of each month at Fanshawe College in London, Ontario. For more information, phone 519-471-1345.

Harry Boyce
Recording Secretary

Gainesville, FL

The Alachua Color Computer User's Group meets the second Tuesday of each month at 7 p.m. at the Operations Center behind the Kelly Power Plant, 555 S.E. 5th Ave., Gainesville, FL.

For more information, contact George McDonald, Rt. 2, Box 530, Alachua, FL 32615, 904-462-5392, or me.

Al Kirk
4617 S.E. 2nd Place
Gainesville, FL 32601
904-377-6285

Northern Virginia

The Northern Virginia Color Computer Club meets monthly in the Community Room of the Manassas Public Library, Manassas, VA. We offer regular classes in Assembly language and Basic, and several members meet informally as an OS-9 SIG.

Anyone wishing more information about upcoming meeting dates or special programs should phone the vice president, Allan Weinstein, at 703-361-2293 (in the Manassas area) or me at 703-820-0658 (in the Washington, DC area).

Logan McMinn
Club Secretary

Northern Illinois

The Northern Illinois Color Computer Club (NICCC) meets on the fourth Tuesday of the month at the Des Plaines Public Library. Contact me for more information.

Richard Ekstrom, Secretary
580 Milton Lane
Hoffman Estates, IL 60194
312-885-2573

Traverse City, MI

Anyone interested in forming a CoCo club in the Traverse City, MI area please contact me.

Richard Nottage
333 S. Garfield Ave.
Traverse City, MI 49684
616-947-0756

CoCo Surveyor

We have developed software for the Color Computer to assist in civil engineering and surveying work, and would like to hear from anyone out there who uses their CoCo for similar applications.

SPACECO
6645 N. Oliphant
Chicago, IL 60631

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

PRICE BREAKTHROUGH

Super Sale on New Disk Drives

Starting at **$169.00!** NEW LOW PRICE

Tandon — Siemens — Remex — MPI — Teac — Shugart — Tabor

40 or 80 Tracks — Single or Dual Head — New 3½'' Drivette™

Our Disk Drives are Capable of Single **and** Dual Density Operation

The **NEWEST** Technology Capable of Operating on Most Popular Computers

Drive a Hard Bargain!!™ For your TRS/80, Color Computer, IBM, Apple, Franklin, MAX/80 5 M.B.-20 M.B. Complete Systems.....from $999.95

Diskette Breakthrough — 10 Pack in Library Case — ~~$18.95~~ NEW LOW PRICE

SAVE!! PLEASE CALL FOR OUR MOST CURRENT PRICE REDUCTIONS.

TOLL FREE ORDERING
1-800-343-8841

GENERAL AND TECHNICAL
1-617-872-9090

Disk Drives (0123) TRS/80-IBM-Apple-TI 99/4A-Franklin-Max/80-LNW ◄
Model I/III/IV Upgrade (Disk Drives-Memory) ◄
Printers—Daisywheel/Dot Matrix ◄
Percom Double Density Controller (Model I) ◄
Color Computer Printer Interfaces ◄
Disk Drive Operating Systems ◄
Repair Services Now Offered—FAST Turn-a-Round ◄
Apple/Franklin Compatible Add-On Drives with Case & Cable ◄
Diskettes in Library Cases ◄

CALL TOLL FREE FOR NEW PRICES

DISK DRIVE CASES AND POWER SUPPLIES ..... starting at $59.95
Printer **Buffers** 8K to 512K ..... starting at $143.95
Holmes Model I/III speed-up Mod ..... starting at $90.00
Cables—Printer/Disk Drive ..... starting at $23.00

Warranty on Disk Drives — 6 Months to 1 Year

SOFTWARE SUPPORT, INC.

One Edgell Road, Framingham, MA 01701 (617) 872-9090

Hours: Mon. thru Fri. 9:30 am to 5:30 (E.S.T.) Sat. 10 am to 4:30 pm

DEALER INQUIRIES INVITED.

TERMS:
M.C./Visa/Amex and personal checks accepted at no extra charge.
C.O.D., please add $3.00.
Shipping: Please call for amount.
Not responsible for typographical errors.

CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular **Tel. (514) 845-1534**
Canadian Toll Free 800-361-5155

Service! Service!

All in stock products are shipped within 24 hours of order. Repair/Warranty service is performed within 24 hours of receipt unless otherwise noted. We accept C.O.D., foreign and APO orders. School and D&B corporate P.O.s accepted.

TRS/80 Registered Trademark Tandy Corp. IBM-PC Registered IBM Corp. Apple Registered Trademark Apple Computer Corp.
Franklin Registered Trademark Franklin Corp. Max/80 Registered Trademark Lobo Int.

DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES DISK DRIVES

trouble with my tapes.

Thanks for a super magazine. I especially like the utility and tutorial programs, since they provide meat for many ideas.

Bert Underwood
Republic of Ireland

German CoCo

The CoCo will do fine in Switzerland. If you use a Swiss TV, you won't get sound unless you use a video interface with an external amplifier and speaker. The 50Hz vs. 60Hz seems negligible.

Modem communication might be a problem, but you can solve it by hardwiring. We should have our first Overseas Colorama BBS running in West Germany by late July 1984.

Peter O. Banz
Ceratec Inc.
Elgin, TX

More German CoCo

I've had no problem with European electrical current. Most European countries use 220V/50Hz current, as opposed to the U.S. standard of 110V/60Hz. A transformer converts the voltage, and the difference between 60Hz and 50Hz hasn't bothered my 64K disk system, which I've been using for two years in Germany.

Donald Keller
APO NY 09114

CoCo by the St. John

The differences in line frequency (50Hz–60Hz) could possibly be a problem, because some devices use line frequency for timing purposes, but I don't know this to be true of any CoCo peripherals.

Of course, an American TV will be useless in Europe except as a monitor, but every CoCo deserves its own screen anyway.

Peter Carr
Fredericton, New Brunswick

Peripheral Buyer's Guide Update

I'm happy to see the April 1984 Peripherals Buyer's Guide (*HOT CoCo*, pp. 44–54), but I must take exception to your claim that Radio Shack's Multi-Pak Interface (#26-3024) does not have buffered circuitry. Only the NMI, HALT, and SLENB lines are not buffered. All others are heavily buffered, and the CTS and SCS lines are independently selected for each slot. This makes interfacing easy and flexible.

John R. Kelty, President
Kelty Engineering

Kelty Engineering

Eds. note—We must also apologize to Mr. Kelty for omitting his Cheap Talker speech-synthesis pack from our April Buyer's Guide.

The Cheap Talker cartridge pack combines with the DEL Software Text Translator to add a voice to your Color Computer. It includes a manager program that lets you save customized dictionaries of often-used words. A complex set of rules built into the translator form words that aren't in the dictionary.

Kelty Engineering recommends using the Cheap Talker with the Multi-Pak Interface and a disk system. The speech synthesis package sells for $69.95 plus $3 shipping and comes with a 16K cassette and a 32K disk and cassette version of the programs. For more information, contact Kelty Engineering, 1440 N 61st, Lincoln, NE 68505, 402-467-3298.

We've Moved

Please note our new address and phone numbers:

Tom Mix Software
4285 Bradford N.E.
Grand Rapids, MI 49506
616-957-0444 (voice)
616-956-9553 (BBS)

No ROMFIX on the J&M

Eds. note—In the June 1984 Dr. ASCII, Ester Horst asked for help using the ROMFIX program ("Disk Utilities," HOT CoCo, September 1983, p. 136) with her J&M disk controller and JDOS. Mr. Esposito suggested that the program should work if she substituted a Radio Shack Color Disk Basic ROM for the JDOS. But we've since learned that the J&M disk controller is not compatible with ROMFIX.

Great Disk Storage

I've found that a plastic ring binder and a supply of loose-leaf plastic envelopes from the stationery store are a great way to file my disks. The transparent envelopes are 6.75 by 8.5 inches and have a lightly textured surface so they don't stick together.

There's enough room in each envelope for two disks, or for a disk and instructions. A small label at the right place on each envelope will let you read the contents of your book when you open it.

A little anti-static fluid helps keep down the dust—but beware of the glossy, sticky envelopes: they are a fatal dust trap.

Colin Stevenson
New South Wales
Australia

Silver CoCo≠White Drive

I recently encountered some unusual problems when I added one of the new white Radio Shack disk drives and disk controller (#26-3029) with the 1.1 Disk Basic ROM to my old silver CoCo (1.1 ROM). The local Radio Shack gave the drive an alignment check and a new chip, but that didn't solve the problem.

Then I learned that the controller's 1.1 Disk ROM won't work with the silver CoCo. I switched it for a 1.0 Disk Basic ROM and now it works fine.

Incidentally, it seems that some of the original CoCo software needs some modification in order to run on a whole new CoCo 2 system.

Sheffield P. Wilds
Pensacola, FL

MC-10 Software Search

Our family owns several MC-10s. Has anyone out there developed software for it? We're looking for database managers, inventory programs, and adventure games.

Patrick J. Terry
401 S. Indiana St.
Warsaw, IN 46580

There's an interesting MC-10 game called Adventure in Bipland from The Dataman, 420 Ferguson Ave. N., Hamilton, Ontario L8L 4Y9, 416-529-1319. Look for a review of it in an upcoming Gameware.

And don't forget about the National MC-10 User's Group, 906-A South Mariana, Tempe, AZ 85281 (Bill Gordons, president). They're publishing an MC-10 software catalog.—eds.

Telewriter-64™

the Color Computer Word Processor

- **3 display formats: 51/64/85 columns × 24 lines**
- ***True* lower case characters**
- **User-friendly full-screen editor**
- **Right justification**
- **Easy hyphenation**
- **Drives any printer**
- **Embedded format and control codes**
- **Runs in 16K, 32K, or 64K**
- **Menu-driven disk and cassette I/O**
- **No hardware modifications required**

THE ORIGINAL

Simply stated, Telewriter is the most powerful word processor you can buy for the TRS-80 Color Computer. The original Telewriter has received rave reviews in every major Color Computer and TRS-80 magazine, as well as enthusiastic praise from thousands of satisfied owners. And rightly so.

The standard Color Computer display of 32 characters by 16 lines without lower case is simply inadequate for serious word processing. The checkerboard letters and tiny lines give you no feel for how your writing looks or reads. Telewriter gives the Color Computer a 51 column by 24 line screen display with *true lower case characters*. So a Telewriter screen looks like a printed page, with a good chunk of text on screen at one time. In fact, more on screen text than you'd get with Apple II, Atari, TI, Vic or TRS-80 Model III.

On top of that, the sophisticated Telewriter full-screen editor is so simple to use, it makes writing fun. With single-letter mnemonic commands, and menu-driven I/O and formatting, Telewriter surpasses all others for user friendliness and pure power.

Telewriter's chain printing feature means that the size of your text is never limited by the amount of memory you have, and Telewriter's advanced cassette handler gives you a powerful word processor without the major additional cost of a disk.

...one of the best programs for the Color Computer I have seen...
— Color Computer News, Jan. 1982

TELEWRITER-64

But now we've added more power to Telewriter. Not just bells and whistles, but major features that give you total control over your writing. We call this new supercharged version Telewriter-64. For two reasons.

64K COMPATIBLE

Telewriter-64 runs fully in any Color Computer — 16K, 32K, or 64K, with or without Extended Basic, with disk or cassette or both. It automatically configures itself to take optimum advantage of all available memory. That means that when you upgrade your memory, the Telewriter-64 text buffer grows accordingly. In a 64K cassette based system, for example, you get about 40K of memory to store text. So you don't need disk or FLEX to put all your 64K to work immediately.

64 COLUMNS (AND 85!)

Besides the original 51 column screen, Telewriter-64 now gives you 2 additional high-density displays: 64 × 24 and 85 × 24!! Both high density modes provide all the standard Telewriter editing capabilities, and you can switch instantly to any of the 3 formats with a single control key command.

The 51 × 24 display is clear and crisp on the screen. The two high density modes are more crowded and less easily readable, but they are perfect for showing you the exact layout of your printed page, *all on the screen at one time*. Compare this with cumbersome "windows" that show you only fragments at a time and don't even allow editing.

RIGHT JUSTIFICATION & HYPHENATION

One outstanding advantage of the full-width screen display is that you can now set the screen width to match the width of your printed page, so that "what you see is what you get." This makes exact alignment of columns possible and it makes hyphenation simple.

Since short lines are the reason for the large spaces often found in standard right justified text, and since hyphenation is the most effective way to eliminate short lines, Telewriter-64 can now promise you some of the best looking right justification you can get on the Color Computer.

FEATURES & SPECIFICATIONS:

Printing and formatting: Drives any printer (LPVII/VIII, DMP-100/200, Epson, Okidata, Centronics, NEC, C. Itoh, Smith-Corona, Terminet, etc).

Embedded control codes give full dynamic access to intelligent printer features like: underlining, subscript, superscript, variable font and type size, dot-graphics, etc.

Dynamic (embedded) format controls for: top, bottom, and left margins; line length, lines per page, line spacing, new page, change page numbering, conditional new page, enable/disable justification.

Menu-driven control of these parameters, as well as: pause at page bottom, page numbering, baud rate (so you can run your printer at top speed), and Epson font. "Typewriter" feature sends typed lines directly to your printer, and Direct mode sends control codes right from the keyboard. Special Epson driver simplifies use with MX-80.

Supports single and multi-line headers and automatic centering. Print or save all or any section of the text buffer. Chain print any number of files from cassette or disk.

File and I/O Features: ASCII format files — create and edit BASIC, Assembly, Pascal, and C programs, Smart Terminal files (for uploading or downloading), even text files from other word processors. Compatible with spelling checkers (like Spell 'n Fix).

Cassette verify command for sure saves. Cassette auto-retry means you type a load command only once no matter where you are in the tape.

Read in, save, partial save, and append files with disk and/or cassette. For disk: print directory with free space to screen or printer, kill and rename files, set default drive. Easily customized to the number of drives in the system.

Editing features: Fast, full-screen editor with wordwrap, block copy, block move, block delete, line delete, global search and replace (or delete), wild card search, fast auto-repeat cursor, fast scrolling, cursor up, down, right, left, begin line, end line, top of text, bottom of text; page forward, page backward, align text, tabs, choice of buff or green background, complete error protection, line counter, word counter, space left, current file name, default drive in effect, set line length on screen.

Insert or delete text anywhere on the screen without changing "modes." This fast "free-form" editor provides maximum ease of use. Everything you do appears immediately on the screen in front of you. Commands require only a single key or a single key plus CLEAR.

...truly a state of the art word processor... outstanding in every respect.
— The RAINBOW, Jan. 1982

PROFESSIONAL WORD PROCESSING

You can no longer afford to be without the power and efficiency word processing brings to everything you write. The TRS-80 Color Computer is the lowest priced micro with the capability for serious word processing. And only Telewriter-64 fully unleashes that capability.

Telewriter-64 costs $49.95 on cassette, $59.95 on disk, and comes complete with over 70 pages of well-written documentation. (The step-by-step tutorial will have your writing with Telewriter-64 in a matter of minutes.)

To order, send check or money order to:

Cognitec
704 N. Nob St.
Del Mar, CA 92014 ↙ 121

Or check your local software store. If you have questions, or would like to order by Visa or Mastercard, call us at (619) 755-1258 (weekdays, 8AM-4PM PST). Dealer inquiries invited.

(Add $2 for shipping. Californians add 6% state tax. Allow 2 weeks for personal checks. Send self-addressed stamped envelope for Telewriter reviews from CCN, RAINBOW, 80-Micro, 80-U.S. Telewriter owners: send SASE or call for information on upgrading to Telewriter-64. Telewriter-compatible spelling checker (Spell 'n Fix) and Smart Terminal program (Colorcom/E) also available. Call or write for more information.)

The Basic Beat

AND THE BEAT GOES ON

by James W. Wood

The saga continues. The Basic Beat was originally an 11-month series designed to teach you the use of Color Basic commands. Since many of you want the column to continue, it will, giving new uses for old commands while still offering lessons for the newcomers.

Presenting short listings to teach computer commands is and has been The Basic Beat's approach. Most listings are useful only for teaching commands, but occasionally a program grows into something you might want to save on cassette or disk.

Each Basic Beat column will consist of a beginner's section and an advanced section for those with some programming experience. All programs that appear will work on any 16K, 32K, or 64K standard or Extended Color Basic machine, and most will work on a 4K CoCo.

Getting Started (Again!)

Writing a program requires a knowledge of Basic commands. You must be aware of the computer's abilities and decide every action you want your program to perform. Whether you are working on a game, an educational program, or a business program, you must decide its purpose before you write it. The computer performs what you program, so once you are proficient at the commands and decide what you want, it is just a matter of taking the time to instruct the machine.

Now, turn on your Color Computer and try running a program. Here are step-by-step instructions for entering Program Listing 1. There are three lines, each beginning with a line number. I numbered the lines by 10. Although this isn't necessary, it leaves room to enter other lines if you decide to modify the program later.

At the end of each line push the enter key. After typing the three lines type LIST and enter. If you have entered the listing correctly, the LIST command will display the entire program on the screen. To make it neater, press the clear key before listing.

Type RUN and press enter. The computer will execute the program lines in order. Line 10 clears the screen. Line 20 prints what is between quotes, and line 30 sends the computer back to line 20, which executes again. The program continues to line 30, which sends it back to line 20.

You are stuck in a loop and can stop the program several ways. You could hold down the shift key and press @ to freeze the action, but the program will continue when you press any other key. To end the program press break. You can replace the sentence in line 20 with any other phrase.

Did you have any problems typing the program? If you typed the wrong letter, back up using the left arrow. Did the letters turn to green-on-black instead of black-on-green? This is Radio Shack's idea of lowercase.

To see inverse lowercase letters hold down shift and press zero. These letters will work within quotes but not when they are used to type commands such as CLS, PRINT, or GOTO. To go back to uppercase press shift and zero together. Type NEW and press enter to erase one program before you enter another.

Program Listing 2 prints whole numbers starting with one and will probably continue longer than you care to watch. Remember how to break the program? After breaking, try restarting with CONT and enter, and then with RUN and enter. CONT will continue the program from where you broke, and RUN starts the program from the beginning.

Line 30 might be mathematically incorrect, but in Basic it means to make A equal to what A used to equal plus one. Experiment with using other numbers intead of 1. You can remove line 20. When you type RUN, it sets the values of all variables equal to zero.

As with most programs, you can improve this one. I don't like the way the numbers are printed in one line down the left hand side of the screen. Retype line 40 as is, but end it with a semicolon. Now when you run Listing 2 the numbers will print across the screen.

Ending a PRINT statement with a semicolon causes the next PRINT state-

```
10 CLS
20 PRINT"WELCOME TO BASIC BEAT"
30 GOTO20
```

Program Listing 1

System Requirements

4K RAM
Color Basic

```
10 A=7
20 A1=3
30 AA=4
40 PRINT"A";"A1";"AA"
50 PRINTA;A1;AA
60 PRINTA+A1;AA-A1;A/AA
```

Program Listing 4

```
10 CLS
20 A=0
30 A=A+1
40 PRINTA
50 GOTO30
```

Program Listing 2

```
10 PRINT"HELLO ";
20 PRINT"THERE ";
30 PRINT"ALL "
40 PRINT"YOU "
50 PRINT"PEOPLE "
```

Program Listing 3

```
10 AAA=5
20 AAB=7
30 PRINTAAA;AAB
```

Program Listing 5

ment to print its message spaced next to the previous printing instead of on the next line. Run Program Listing 3 for another demonstration of ending PRINT statements with and without semicolons.

Program Listing 4 reviews PRINT statements and variables. Line 40 prints the letters A, A1, and AA because they are in quotes. They are printed close together because of the semicolons. Line 50 prints the values of A, A1, and AA because there are no quotes. Notice that when you use semicolons, words or letters have no spaces between them while numbers have a separating space. Line 60 demonstrates Basic's ability to perform mathematical operations within a PRINT statement.

Variables, the X and Y unknowns from math class, are expressed in many ways on the computer. Program Listing 4 shows that variables can be one letter, two letters, or one letter followed by a one digit number. They can be longer than two letters, but this can cause problems as Program Listing 5 verifies. The computer confuses AAA with BBB. In line 30, they both equal 7.

Although you don't have too many commands yet, next month you will begin to build your Basic vocabulary and create some more interesting programs.

For the Advanced Student

I need to correct a statement made in an earlier issue. I wrote that it was difficult or impossible to use the four arrow keys with the INKEY$ command. I goofed. It is true that pushing the left-, right-, and down-arrow keys produces no typing on the screen, so it is impossible to have a line such as A$ = "B" using a down-arrow key instead of the B.

Look at Program Listing 6, a simple drawing program, to discover the secret. In line 20 you set D$ (for down) equal to CHR$(94). Pressing the down-arrow key doesn't register on the screen, but the computer knows when you do, and the character CHR$(94) is sent to memory.

The rest of line 20, and line 30 define the codes for the other arrow keys. Line 40 uses INKEY$ to scan the keyboard. Usually INKEY$ must have a key released to be read more than once. Line 50 fools the computer into thinking any key on the keyboard that is held down was also released. Lines 60–130 check for which arrow key is pressed and make sure your drawing can't go off the screen causing a function call (FC) error. Line 140 sets a graphics position and returns to line 40 to check again for pressed arrow keys.

Program Listing 7 is my contribution to your collection of useful programs. It also demonstrates nested FOR loops, arrays, and string manipulation. The program will change the last four digits in your phone number to all the possible words that the telephone letters could produce. There are no letters for the numbers zero and one. The phone company made that rule, I didn't. I found out that my place of employment's phone number translates to BACO. The program displays all 81 possibilities on the screen without scrolling any off. ■

Address correspondence to James W. Wood, 424 N. Missouri, Box 507, Atwood, IL 61913.

```
10 CLS0:X=31:Y=15
20 U$=CHR$(94):D$=CHR$(10)
30 L$=CHR$(8):R$=CHR$(9)
40 A$=INKEY$
50 FOR N=338TO345:POKE N,255:NEX
T N
60 IF A$=U$ THEN Y=Y-1
70 IF A$=D$ THEN Y=Y+1
80 IF A$=R$ THEN X=X+1
90 IF A$=L$ THEN X=X-1
100 IF X<0 THEN X=0
110 IF X>63 THEN X=63
120 IF Y<0 THEN Y=0
130 IF Y>31 THEN Y=31
140 SET(X,Y,8):GOTO40
```

Program Listing 6

```
5 CLS
10 PRINT"IF THE LAST FOUR DIGITS
 OF"
20 PRINT"YOUR PHONE NUMBER DO NO
T"
30 PRINT"CONTAIN 1 OR 0, I WILL
TRY"
40 PRINT"TO FIND A WORD TO HELP
YOU"
50 PRINT"REMEMBER THE NUMBER."
60 PRINT"WHAT ARE THE LAST FOUR
DIGITS"
70 INPUT PN$
80 FORA=1 TO 4:N(A)=VAL(MID$(PN$
,A,1)):NEXTA
90 FOR A=2 TO 9:READ L$(A):NEXT
A
100 DATA ABC,DEF,GHI,JKL,MNO,PRS
,TUV,WXY
110 FOR A=1 TO 3:FOR B=1 TO 3
120 FOR C=1 TO 3:FOR D=1 TO 3
130 PRINTMID$(L$(N(1)),A,1)+MID$
(L$(N(2)),B,1)+MID$(L$(N(3)),C,1
)+MID$(L$(N(4)),D,1)+" ";
135 W=W+1:IF W=6 THEN PRINT:W=0
140 NEXTD,C,B,A
```

Program Listing 7

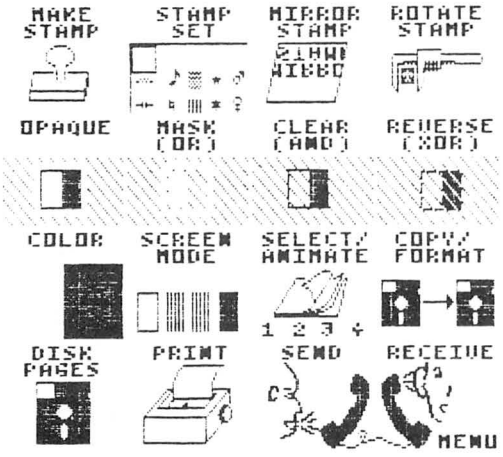
See List of Advertisers on page 99

GRAPHICOM HEADQUARTERS U.S.A.

Simply stated - the finest graphics program written for the COCO (or any other computer)!

FEATURES

- U-S-E-R F-R-I-E-N-D-L-Y !
- 4 Mode (Including Hi-Res Artifact)
- Animate Mode
- Color Palate with over 15 color patterns for use with Hi-Res Artifact
- Send/Receive pictures over standard modem at 300, 600 or 1200 baud
- Supplied utility allows capturing Hi-Res Screens from most COCO arcade type games (Even protected ones)
- Multiple Hi-Res character fonts (user changable & definable)
- Supplied utility for transferring graphicom screens to basic or other M/L Programs
- Supplied utility for loading screens from basic or other sources
- Built in Screen print (Pre-defined for Epson, C-Itoh, RS LP VII, LP VIII, DMP 100, DMP 200, DMP 120, GCP 115, GEMINI 10 and OKI) 110 to 9600 baud
- Slow scan television send/receive options
- Many additional features, operating hints, hardware mods and suggestions, etc. etc.!



★ **EASY TO LEARN GRAPHIC MENU** ★

REQUIRES 64K COCO - 1 DRIVE - JOYSTICKS

ON DISK $24.95

G.C.U.

GRAPHICOM UTILITY

FEATURES...

*** MULTI DRIVE - COPY PICTURE FROM ONE DISK TO ANOTHER.
*** KILL (BLANK OUT) INDIVIDUAL PICTURES ON A PIX DISK.
*** EASILY LOAD BINARY FILE TO PICTURE LOCATION.
*** DISPLAY INDIVIDUAL PICTURE.

ALL FOR $19.95

COMPUTIZE INC.
P.O. BOX 207
LANGHORNE, PA 19047

Check or M.O.
VISA MasterCard
Add $2.00 Shipping
PA Res. add 6% sales tax

215-946-7260

181

TELE-FORM

Telewriter-64 Disk & Tape File Formatter

- **Formats text—from within Telewriter buffer—to disk or tape file as Telewriter print formatter does.**
- **Re-load file to see centering, margins, page numbering, headers, justification, etc.**
- **Helps integrate Telewriter with Data Base Managers for direct mail use.**
- **Send formatted manuscripts electronically with page numbers, headers, centering, etc.**

$29.95—DISK $24.95—TAPE
Add $2.00 Shipping

Cigna Co., 115 Belmont Street
Rochester, New York 14620

(Telewriter-64 and Telewriter are Trade Marks of Cognitec)

219

CP/M

FOR YOUR COLOR COMPUTER

- Now have access to the largest library of programs available
- CP/M is the recognized leader in professional and personal software
- Adds the power of a 4 MHz, Z80A
- CP/M 2.2 included
- No modifications are necessary. Simply plug into the cartridge port and plug Radio Shack's disk controller into it.
- Maintains full Radio Shack compatibility
- Requires 64K memory and one or more disks with controller

$250.00

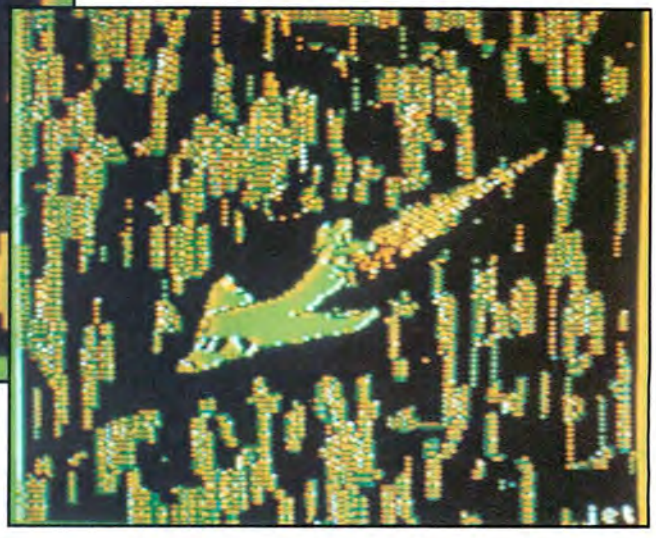
WAYNE TECHNOLOGY

P.O. BOX 5196 • ANAHEIM, CA 92804-1196
(714) 772-5757

170

Radio Shack™ Tandy / Radio Shack Corp. / CPM™ Digital Research

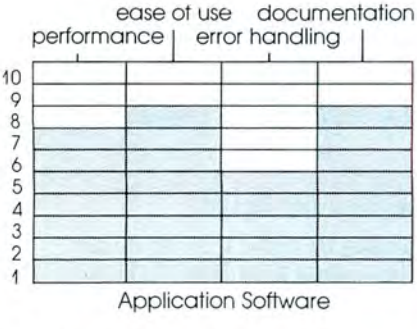
BY GUIER S. WRIGHT



GRAPHICOM:

GRAPHICS EDITOR EXTRAORDINAIRE

Graphicom is to the graphics designer what a good word processor is to a professional writer.



Graphicom
Cheshire Cat Computer Creations
P.O. Box 115
Lafayette, CA 94594
64K, 2 joysticks
$30 disk

Graphicom is a screen-oriented, joystick-controlled, graphics-editing and developmental software tool for the 64K Color Computer. It gives you features such as stamp making, telecommunication abilities, faster disk I/O, simple animation, and numerous other options that you won't find on other graphics screen-editing programs.

Since Graphicom's creators wish to remain anonymous, for whatever reasons, I drew my own mental image of how they developed the program. Perhaps an architect or an electrical engineer and a programmer sat around talking about the frustrations of their jobs. There were just so many little things that got in the way of their creative flow.

The engineer said that he spent far too many hours with a plastic template and ruler drawing resistor, switch, and

Now you can learn how to use your color computer for more than just games . . . with HOT CoCo magazine.

With the right information on programming utilities, debugging, and graphics there's no limit to what you can do with your color computer. **HOT CoCo** gives you that information. It can make your computer a versatile tool that you'll find indispensible. **HOT CoCo** is packed with:

- **Business application instructions** to help you understand what the color computer can do at the office. You can use these applications immediately because they're written in plain english.
- **Home management help**—let **HOT CoCo** show you how everyday chores can be done on your machine. You'll be surprised at just what you can do and just how much time can be saved with your color computer.
- **Programming tips & tutorials**—**HOT CoCo** will show you how to program. It's loaded with programming techniques and hints to help the novice and expert programmer write and improve their programs.
- **New product reviews & announcements**—if you're looking for equipment to expand the use of your computer, **HOT CoCo** reviews numerous hardware and software products each month. Plus, **HOT CoCo**'s new product announcements let you comparison shop at home—spend more time at your computer and less time in computer stores.

And **HOT CoCo** is loaded with challenging games to provide hours of fun and excitement for your whole family.

Let **HOT CoCo** show you how much time you can save with your color computer. Order **HOT CoCo** today!

Take advantage of this money-saving offer. Get 12 issues of **HOT CoCo** for only $24.97. A 13th issue is yours FREE with pre-payment (check or credit card). Use the attached order card, the coupon, or call

TOLL FREE 1-800-258-5473.
IN NH CALL 1-924-9471.

☐ ***YES! I want more use from my CoCo!***

Send me 12 issues of **HOT CoCo** for $24.97 now! I understand that with payment enclosed or credit card order I will receive a FREE issue making a total of 13 issues for $24.97.

☐ *Check Enclosed* ☐ *MC* ☐ *VISA* ☐ *AE* ☐ *Bill Me $24.97 for 12 issues*

Card #_____ Exp. Date_____

Signature_____

Name_____

Address_____

City_____ State_____ Zip_____

Canada & Mexico $27.97, 1 year only, US funds drawn on US bank.
Foreign Surface $44.97, 1 year only, US funds drawn on US bank.
Foreign airmail please inquire. Please allow 6–8 weeks for delivery.

HOT CoCo • PO Box 975 • Farmingdale, NY 11737

347F4

transistor symbols. He would do a rough drawing and then spend the next two days cleaning it up.

The programmer said that he thought flowcharting was a good idea, but who could ever spend the time drawing all those little decision, output, and logic boxes? Authors had word processors, so why wasn't there something for the designer? The engineer? The people who do their creating with drawings, charts, and diagrams?

Of course, there were CAD (computer-aided design) systems, but these were expensive, highly specialized, and difficult to learn. To develop something along those lines could take a great deal of money, and it wasn't exactly what they were looking for. They needed a program that was easy to use, inexpensive, and flexible enough to handle many different design problems. They needed something like Graphicom.

Graphicom's designers must have begun by deciding exactly how the finished program should appear on the screen, rather than what they wanted it to do, or how they should program it.

When they knew how it should look, they set about deciding what it should do for them. It had to be easy to use, and Graphicom is easy to use. It had to be very flexible, and Graphicom is extremely flexible. It had to be open-ended, in that it could and would be improved, and in the few months it's been available, both authors and users have made changes.

But perhaps what really sets Graphicom apart from other software is the effort its authors have made to make it available to anyone who wants it. The program is not copy-protected and certainly isn't expensive—Cheshire Cat has even made the source code available to those who might use it, though the main program is written in Forth.

Ease of Use

The program, while not perfect in every respect, is an excellent tool for creating. It is to designers, programmers, software developers, or anyone who wants to use the graphic capabilities of the Color Computer what a good word processor is to a writer.

You perform almost all the creating via two joysticks, which is one of the few difficulties in the program. The instruction manual suggests that you attach both joysticks side-by-side on a solid surface, and I recommend this system. It also suggests using one of the joysticks as a foot pedal, since you use only the fire button on one to flip back and forth between screens.

The system sometimes gets a bit awkward when you try to hold down both fire buttons while using the joystick to move the cursor around the screen. This is only a minor problem, however, and you'll probably find yourself more frustrated with the lack of sensitivity in certain joysticks, especially as you work on detailed sections of a drawing.

It was only a matter of 10 or 15 minutes from the time I first loaded Graphicom until I was working on those detailed sections. Graphicom is extremely easy to learn and use. Marty Goodman, the author of the 32-page manual, admits that the program is not nearly as complex as most other software, but it is not so simple that you can throw away the documentation.

There are some aspects of the program that are not instantly obvious, and the manual, while containing a handful of typos, is a good set of instructions. It is clearly written, unintimidating, and even fun at times, but in no way condescending or overly simplistic. In most instances, it offers both technical and nontechnical explanations for terms and options.

"It is like a graphics word processor, and should certainly help the graphics programmer."

Graphicom's Features

The stamp functions are one of the most important advantages over other graphic editors. You can use the make-stamp option to make a drawing or symbol as a stamp that you can save or use right away. The program also includes its own set of predefined stamps such as flowchart or electronic symbols.

Once you've selected or defined a stamp, you can use and manipulate it in many ways. Not only can you rotate the stamp 90 degrees at a time, you can also reverse the image using the mirror function. When you have the proper orientation, you can select any of the following ways to apply the stamp to one of the four user screens:

- The opaque mode is like placing a postage stamp over a part of the picture, completely covering anything underneath.
- The clear mode is similar to a rubber stamp—the lines of the stamp obscure anything underneath.
- In the reverse mode, the stamp doesn't put any lines over the drawing. In fact, it erases that part of the drawing where the line would be.
- The mask mode does the reverse of clear. It obscures all of the drawing except where the lines of the stamp are.

As with many aspects of Graphicom, the results are easier to see than to describe, but as the manual says, once you get used to the power of these options, you might wonder how you got along without them.

Of course, you can change color sets and screen resolution, and drawing is a simple matter of pressing the fire button to "tack" down one end of a rubber-band-like line and then moving the joystick to put the line where you want. Then press the fire button again to place it.

Graphicom comes with a screen-dump option that will work with a number of printers. You can also mix text and graphics on the screen. The program offers a variety of type styles, or you can design your own characters.

One of Graphicom's most interesting features is its ability to send pictures over a modem. Included with the program is a utility designed by Mike Ward that lets you transmit a Graphicom-designed picture to any other computer, and the receiver doesn't need Graphicom in order to see the results. Using this unique utility, a number of people have already started compiling a library of pictures on the CompuServe Color Computer SIG (Go PCS 126).

Another Graphicom utility lets you save a picture in normal binary format and load and use it in any Color Computer, with or without Graphicom. Or, you can transform any picture in the Color Computer's memory into Graphicom format so you can edit, save, or print it.

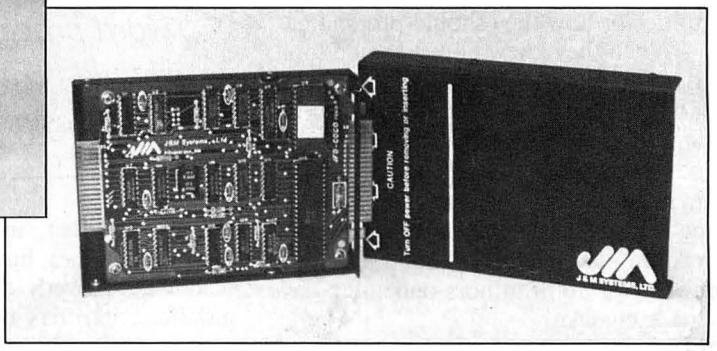
There are faults with Graphicom, but to judge it on the basis of other commercial software is to miss the point. It is like a graphics word processor, and should certainly help the graphics programmer, designer, or doodler in that it offers easy access to the Color Computer's picture-making capabilities.

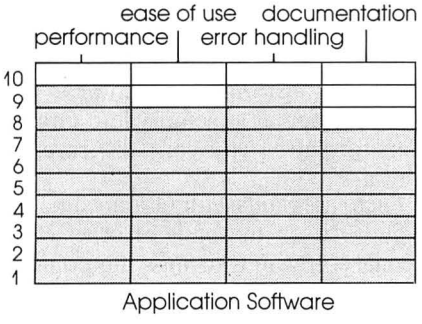
It is also a vehicle for communicating ideas that don't fit easily into words, and that is where it steps beyond other graphics editors: Graphicom lets you do more than just draw pictures; it helps you manipulate and depict your ideas, and then communicate them.

As it is, Graphicom is a great program—and it has provided a basis for something even better. All that for $30. ■

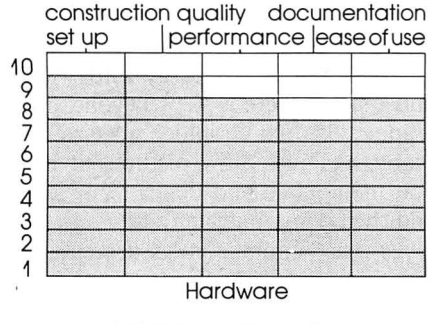
BY HOWARD CULBRETH

A DISK SYSTEM OF A DIFFERENT DOS





JDOS Disk Basic



J&M Disk Controller

Looking for a reliable disk drive at a reasonable cost? J&M's drive with JDOS might be it.

JDOS Disk Basic
J&M Disk Controller
J&M Systems Ltd.
137 Utah N.E.
Albuquerque, NM 87108
505-265-1501
16K, Extended Color Basic
$49, JDOS Disk Basic
$149, J&M Disk Controller

J&M Systems Ltd. has an established reputation for its high-quality products and reliable support. Recently, they have extended their expertise with a disk controller and their JDOS disk-operating system for the Color Computer.

The Controller

The disk controller comes in an attractive black aluminum case, has gold edge connectors, and uses a digital phase-lock loop and precompensation circuit. The circuitry eliminates the adjustments necessary with a Tandy controller, and the edge connectors eliminate the periodic cleaning routine required for the Tandy unit.

The J&M controller is hardware compatible with Tandy's unit, and since it does not require a 12-volt power supply, you can use it with the original CoCo and the Color Computer 2. This controller comes complete and assembled and requires no modifications to either the computer or the controller. Plug it in, power up and it's ready to use.

You can use J&M JDOS with their controller or, subject to availability, you can opt for Tandy's Disk Basic. J&M currently provides Disk ROM version 1.1 to those customers who prefer

to purchase the Tandy DOS.

J&M has had trouble getting the ROMs from Tandy, and you don't get a manual if you select the Tandy DOS. If, however, you select JDOS, you'll get J&M's reference manual.

JDOS

JDOS Disk Basic is an EPROM that plugs into your disk controller, taking the place of Tandy's Disk Basic ROM. You can also install it in Tandy's disk controller. It offers all the functions and commands of the Tandy ROM, plus several additional ones, and comes with a comprehensive reference manual.

JDOS is a bargain. The software needed to give you JDOS's extra functions would probably cost more than this operating system, and JDOS's functions are available as soon as you turn on your disk system. You don't need to find your utility disk, reserve memory, and load and execute another program.

Besides the commands that come with the Tandy DOS, JDOS automatically numbers lines in Basic programming. It also loads and executes the OS-9 and Flex operating systems from drive 0, and offers a function to trap errors and maintain control over the program currently executing. It tells you the line number and error condition—a most valuable feature.

You also get options to set the step rate that the controller uses for the disk drives and to switch the computer into the all-RAM mode, copy the ROMs into RAM memory, and continue execution of the ROM code.

There are, however, some differences in syntax between JDOS and the Tandy DOS. For example, JDOS stores the particular format of a given disk in RAM. Any directory-access operation updates this format. Therefore, if you use JDOS's DSKI$ or DSKO$ operations, your information must be current and accurate.

Also, the Free command displays the number of free sectors remaining on the disk, not the number of free granules as the Tandy DOS displays. For the number of free granules, divide the result of the JDOS Free command by nine for single-sided disks, and by 18 for double-sided disks. Even if you're familiar with Tandy DOS's operation, consult the JDOS operating manual to make sure you understand all commands.

Hardware Considerations

As now configured, JDOS handles any combination of up to three 35- or 40-track, single- or double-sided, double-density disk drives. Even so, you can still read and write Tandy-compatible disks.

"J&M has made a bona fide commitment to JDOS and intends to offer strong and reliable support. They've shipped me two upgrades and a revised reference manual for my system, and it has only cost me postage to return the EPROM to J&M."

Tandy uses pin 32 on their cables to select drive 3 (the fourth disk drive, if you're using that many), because they don't sell a double-sided drive. Other drive manufacturers, however, use this pin for the side-select signal on their double-headed drives. Because the disk controllers do not provide a separate side-select signal, the software (DOS) provides that signal on pin 32 to communicate a side select to the disk drive.

Therefore, JDOS can support only three drives, because it uses pin 32 for side select, precluding its use for drive 3.

Disk Capacity

The Tandy DOS has a 68-file-name directory capacity (the total number of file names that a disk can contain). JDOS expands this capacity to 72 when used with a 40-track drive and a disk formatted to 40 tracks.

Although it gives you a substantial increase in storage capacity under JDOS with 40-track, double-sided drives, remember that you are still limited to three drives. If you use only single-sided drives, this limitation will keep your total system capacity to less than that of a four-drive Tandy system.

Hardware Compatibility

With JDOS you can connect standard, double-sided drives to the system and start operating immediately. Previously, under the Tandy DOS, it was necessary to modify the double-sided drives with a 74LS86 chip and configure each side as a separate physical drive.

Software Compatibility

It is difficult to define software compatibility. Under Basic, compatibility is of little concern, except for the side effects of some of the commands and the manner in which programmers might use them.

Both Basic and machine-language programs that do not use the documented and recommended ROM entry point often crash or execute unreliably. JDOS's recommended disk ROM entry point, DSKCON, is at the same address as the Tandy DOS. The JDOS manual provides several examples of machine-language routines to allow communication with the disk I/O driver.

Not all programmers, however, abide by these guidelines. There are many fine programs that don't work smoothly with Tandy's 1.1 disk ROM. In those same cases, you can expect similar compatibility problems with JDOS. You pay a heavy price for using illegal ROM calls. Software that provides its own DOS should perform without alteration.

Support

J&M has made a bona fide commitment to JDOS and intends to offer strong and reliable support. They've shipped me two upgrades and a revised reference manual for my system, and it has only cost me postage to return the EPROM to J&M. I don't expect that they can continue this policy forever, but it is refreshing.

If past performance is any indication of what to expect in the future, you don't have to worry about J&M's support policies.

Although there are software and hardware requirements and limitations to consider, it is difficult to find a better bargain than JDOS. While not the answer to everyone's needs, it enhances the CoCo's capabilities in fine fashion.

Likewise, the J&M Disk Controller is a reasonably priced, reliable performer. Don't discard your Tandy unit to buy this one, but it is worthy of consideration when you decide to purchase your next one. And it does give you a chance to buy disk drives of your own choice: 35-track, 40-track, single- or double-sided. ■

Address correspondence to Howard B. Culbreth, 419 Mount Vernon Drive, Tabb, VA 23602.

BY JAMES J. BARBARELLO

ROM HACKER-

This project, based on the 6821 PIA, lets you interface an IC tester, robot arm, and a Logo turtle-like robot.

In the past, I have given hardware enthusiasts some interesting, simple projects. In response, you called, "More!" Apparently, the fascination of an interesting, low-cost hardware project is, to the true hardware hacker, irresistible. Well, fellow hackers, in this series, I'll present a number of new and exciting hardware projects for the CoCo.

I call this series the ROM Hacker because you will interface all projects to the CoCo through its cartridge (ROM) port. In Part I, I'll show you how to build a master interface using a 6821 peripheral interface adapter (PIA). Through this interface you'll connect a series of different, exciting projects. (Did I mention a homebrew robot arm? More on that later.) You've got a lot of work ahead of you, so let's get started.

The PIA

The PIA is a 40-pin integrated circuit used for universal interfacing of peripheral equipment to the 6800 series of microprocessors (including the CoCo's 6809 MPU). It has two 8-bit bidirectional peripheral data buses and four control lines (which you won't be using). Most importantly, the PIA does not require any significant external circuitry to interface to low-power peripheral hardware.

The PIA is no stranger to the CoCo; there are two inside it already. You use them for such tasks as interfacing the keyboard and cassette deck, and for RS-232, joystick, and sound functions. To the 6809, the PIA appears as another memory device; you can use Basic's PEEK and POKE functions to read and write to it.

I said that the PIA has two 8-bit data buses (A and B). For each bus, there are three registers: the peripheral register, the data-direction register, and the control register. You send and receive data through the peripheral register. You use the data-direction register to tell the PIA to set up each data line as an input or an output. You use the control register to select either the peripheral or data-direction registers.

To better understand this concept, consider an analogy. You want to order some transistors and return a transformer to a mail-order firm. When you call the company, the receptionist answers the phone. Based on your conversation, she determines your needs and directs you to a salesperson. The salesperson sets everything up, taking your order, giving you a return authorization number and then alerting the shipping department. The shipping department does the actual work of shipping your order.

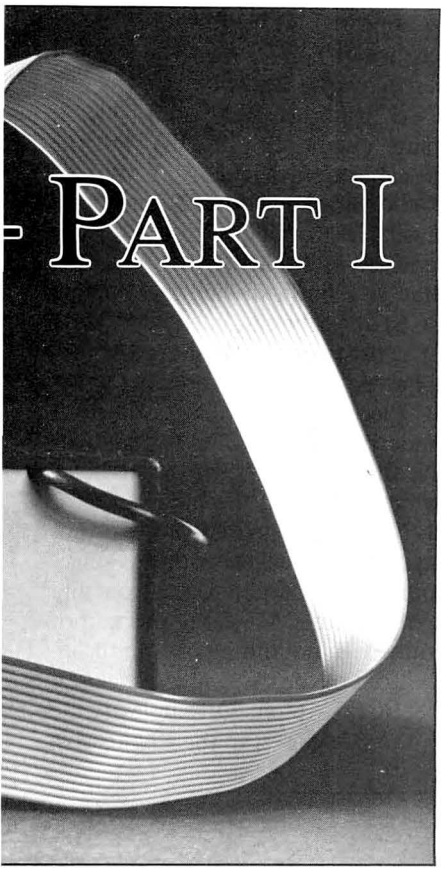
Consider the order and return as data input and output. The receptionist is the control register, guiding you to the salesperson (whom you can think of as the data-direction register). It's the salesperson who tells the shipping department (peripheral register) what to do. But it's the shipping department that does the real work (data transfer).

By sending data to the proper address, you can set up the PIA to have its peripheral lines all inputting data, all outputting data, or any combination of input and output. You must connect both address and data lines to the PIA, and provide some other signals to it. Luckily, they are all available at the ROM port.

You need one more piece of information before you can set up a functional interface. Remember that you can PEEK and POKE the PIA like any other memory device. But how do you know where to PEEK and POKE to? And how does the PIA know when we are addressing it and no other memory location?

System Requirements

16K RAM
Extended Color Basic



By connecting the address lines properly, you can give the PIA a distinct address. The area of memory starting at $C000 (C000 hex, or 49152 decimal) is already reserved for the ROM port. If you use the master interface without any other ROM-port devices (such as a disk drive), you need only connect five address lines to it.

Connecting a PIA as shown in Fig. 1 sets its address at $C000 through $C003. Side A of the PIA is addressed at $C000 and $C001, while side B is addressed at $C002 and $C003.

Programming the PIA

Based on the physical connections of Fig. 1, Table 1 shows how you access the PIA's different registers. Address lines A1 and A0 are connected to register-select lines RS0 and RS1 respectively. Concentrate on side A first. A POKE &HC001,0 makes all bits in control register (CR) A zero (including, of course, bit 2). This switches in the data-direction register (DDR), and switches out the peripheral register (PR).

Now, when you send data to address &HC000, it is stored in the DDR and used to determine which PR lines are inputs, and which are outputs. For instance, if you send 255 to the DDR (i.e., all ones), all PR lines will be configured as outputs. If you send 170, the eight PR lines will alternate (out, in, out, in, etc.).

With this done, you must switch out the DDR and switch in the PR, because actual data transfer between the PIA and a peripheral device is by way of the PR lines. You again address CR A, this time setting bit 2 (POKE &HC001,4).

For side B, you need only to add two to the above addresses. For instance, you address CR B at $C003 and the PR B and DDR B at addresses $C002. Once you have configured the PIA, you send data (POKE) and retrieve data (PEEK) at the lower address ($C000 for side A, and $C002 for side B).

Sending a one to an output line forces it high; sending a zero forces it low. If an input line returns a one, it was high when you looked at it; if it returns a zero, it was low when you looked at it. When you send a zero or one to an output line, it stays in that state until you change it. When you PEEK an input line, you get its status at that instant. It could change to the opposite state immediately after you PEEK it, and you wouldn't know unless you PEEK it again.

Programming the PIA in Basic

As the first example, suppose you want to program the first four lines of side A as inputs, the second four lines of side A as outputs, and all eight lines of side B as outputs. Further, you want the first two output lines on side A to be low, the second two to be high, the first four output lines on side B to be high and the last four output lines on side B to be low. This two-line Basic program does that for us:

```
10 POKE &HC001,0:POKE &HC000,240:
   POKE &HC001,4:POKE &HC000,192
20 POKE &HC003,0:POKE &HC002,255:
   POKE &HC003,4:POKE &HC002,15
```

Looking at line 10 first, POKE &HC001,0 loads CR A with 0, selecting DDR A. Then POKE &HC000,240 sets lines 0, 1, 2, and 3 as inputs and lines 4, 5, 6, and 7 as outputs (240 decimal is 11110000 binary with the bits proceeding from most significant to least significant). Thus ones are sent to DDR A bits 7, 6, 5, and 4, and zeros to DDR A bits 3, 2, 1, and 0. Now, POKE &HC001,4 loads bit 2 of CR A with a 1, since 4 decimal = 00000100 binary. This selects PR A.

Finally, POKE &HC000,192 sets PR A lines 5 and 6 as low and 7 and 8 as high (192 decimal = 11000000 binary). Line 20 performs a similar function for side B. It sets all lines to outputs by placing a one in each bit of DDR B (POKE &HC000,255), and then sets the first four lines high and the last four lines low (POKE &HC000,15). If you add

```
30 PRINT PEEK(&HC000),PEEK(&HC002):
   GOTO 30
```

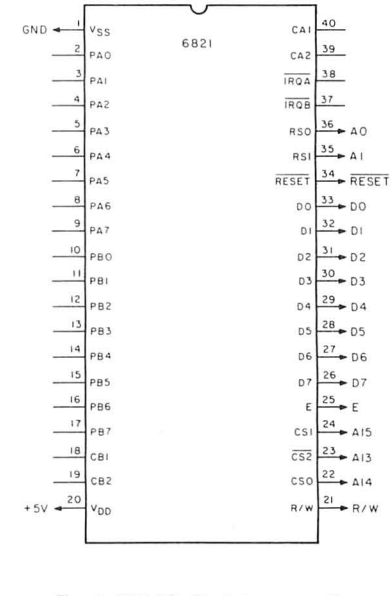


Fig. 1. PIA/CoCo Interconnections

Address		Control Register			
A1	A0	A-Bit 2	B-Bit 2	Register Addressed	POKE
0	1	don't care	don't care	Control register A	$C001,0
0	0	0	don't care	Data-direction register A	$C000,x
0	0	1	don't care	Peripheral register A	$C001,4
1	1	don't care	don't care	Control register B	$C003,0
1	0	don't care	0	Data-direction register B	$C002,x
1	0	don't care	1	Peripheral register B	$C003,4

Table 1. PIA Addressing for Register Access

you can continually view the status of the PIA.

To know which lines are high and low, you must convert the numbers that are returned into their binary representation. The binary bits in the converted numbers correspond to the PIA lines (bit 0 corresponds to PR line 0, bit 1 to PR line 1, and so on). You can accomplish this easily in Basic programming, and this first ROM Hacker project includes such Basic code.

Now that you have some of the theory under your collective belts, you can get on with the task of constructing the Master Interface, without which you can't begin any of the projects I have planned.

The Master Interface

Figure 2 shows the schematic for the Master Interface. It consists of a single PIA and input/output cables, allowing easy connection, and removal from the CoCo and my future projects. For low-power projects, you can use the CoCo's 5-volt supply directly. For projects requiring higher power levels, you can use the power cable as a ground reference and supply the needed current from a separate power supply.

The 40-pin cable plugs directly into the CoCo's ROM port. The 16-pin DIP (dual inline package) jumper cable lets you connect the 16 PIA lines to any external device by way of a single DIP socket on that project's printed circuit board (PCB). The master interface uses readily available components and simple construction techniques.

Construction

If you are new to PCB construction, refer to my "PC Board Primer" series which appeared in the July and August 1983 issues of *HOT CoCo* (pp. 60 and 52, respectively). Begin by constructing the PCB from the PC layout of Fig. 3. All parts except for the PIA and the 40-pin cable are available from your local Radio Shack. (The PIA and cable are available from Spectrum Projects, as identified in Table 2.)

After you finish drilling and etching the PCB, open the case cited in the parts list. It consists of two U-shaped aluminum halves, one black and the other white. Place the four adhesive feet on the underside corners of the white half. Attach the L brackets on the inside of the white half using the hardware provided. Now place the PCB, foil side up, between the L brackets with the longer side of the PCB parallel to the front and back of the case. Mark the position of the four PCB mounting holes on the inside bottom of the case. The side closest to the PCB's 16-pin DIP holes is the front; the side closest to the 40-pin cable

```
1 REM** INTTEST
2 REM** V1.0 - 2/5/84
3 REM** J.J. BARBARELLO
4 REM
10 CLS:DIM B(16),P(16),V(16)
20 PRINT"  MASTER INTERFACE OUTP
UT TEST":PRINTSTRING$(32,131)
30 POKE&HC001,0:POKE&HC000,255:P
OKE&HC001,4:POKE&HC000,0
40 POKE&HC003,0:POKE&HC002,255:P
OKE&HC003,4:POKE&HC002,0
50 P=147:FORI=0TO7:P(I+1)=P+I*32
:NEXT
60 P=362:FORI=9TO16:P(I)=362-(I-
9)*32:NEXT
70 FORI=0TO7:B(8-I)=INT(2^I):B(1
6-I)=B(8-I):NEXT
80 A$=STRING$(13,32)+CHR$(129)+S
TRING$(4,128)+CHR$(130)
90 PRINT@68,"pa=  0   SOCKET   p
b=  0"
100 PRINTTAB(13)STRING$(6,140);S
TRING$(12,140)
110 FORI=1TO8:PRINTA$:NEXT
120 PRINTTAB(13);STRING$(6,128);
STRING$(12,140)
130 A$="CABLE":FORI=1TO5:PRINT@1
57+32*I,MID$(A$,I,1);:NEXT
140 FORI=1TO8:PRINT@P(I),USING"
  [#]";I;:NEXT:FORI=9TO16:PRINT@
P(I)-4,USING"[##]";I;:NEXT
150 FORI=1TO16:PRINT@P(I),V(I);:
NEXT
160 PRINT@448,STRING$(63,32);:PR
INT@448,;
170 INPUT"TOGGLE WHICH PIN (1-16
)";V
180 IF V<1 OR V>16 THEN 160
190 V(V)=(V(V)=0)*-1:PA=0:PB=0
200 FORI=1TO8:PB=PB+B(I)*V(I):NE
XT
210 FORI=9TO16:PA=PA+B(I)*V(I):N
EXT
220 PRINT@71,USING"###";PA;:PRIN
T@89,USING"###";PB;
230 POKE&HC000,PA:POKE&HC002,PB
240 IF PEEK(&HC000)=PA AND PEEK(
&HC002)=PB THEN 150
250 PRINT@64," ":FORI=1TO100:NEX
T
260 PRINT@77,"error":FORI=1TO100
:NEXT:GOTO250
```

Program Listing. Master Interface Tester

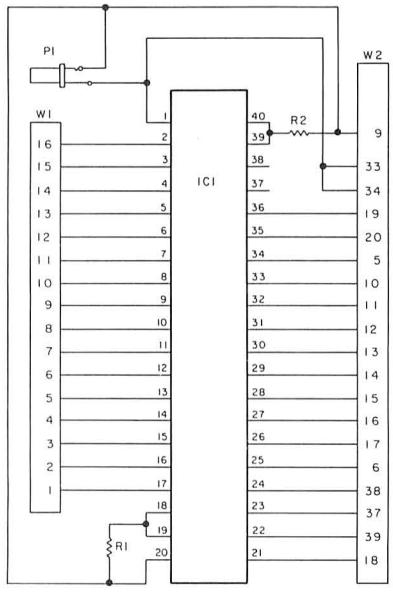


Fig. 2. Master Interface Schematic Diagram

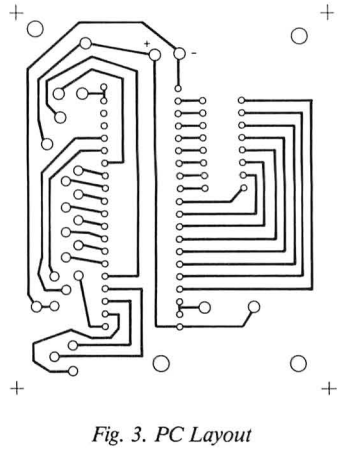


Fig. 3. PC Layout

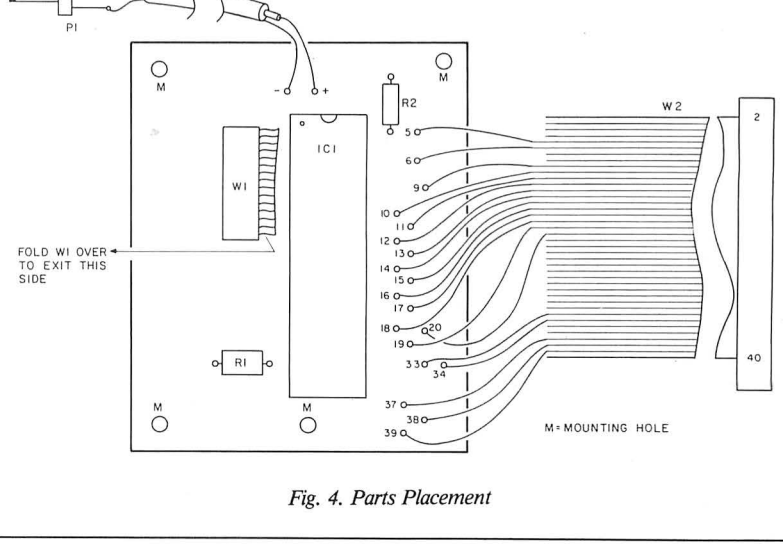


Fig. 4. Parts Placement

↙ See List of Advertisers on page 99

holes is the back. Remove the PCB and drill the four mounting holes with a 1/8-inch bit.

Set the PCB and case aside and begin preparing the 40-pin cable. Place the cable in front of you with the connector to your left and pin 1 (identified on the connector) farthest from you. Call this the top. The free cable end will now be on your right, with lead 1 also on the top.

Using a pencil, make a line 1 inch in from and parallel to the free end of the cable. Now, using a sharp hobby knife, begin separating each of the 40 wires in the cable. To do this, place the tip of the knife in the depression between the first and second lead at the pencil line you drew previously. Pierce the cable and bring the knife toward the free end of the cable. You will now have the first lead free from the cable. Repeat this procedure for the remaining 39 leads.

Only 19 of the 40 leads are required. Using a pair of cutting pliers, snip off the following 21 leads at the pencil line:

1,2,3,4,7,8,21,22,23,24,25,26,27,28,29,30,31, 32,35,36,40

The final preparation step is to carefully strip 1/4 inch of insulation from each of the remaining 19 leads. Do this very carefully, twisting the exposed strands of each lead together.

With the 40-pin cable prepared, return to the PCB. Solder the 40-pin DIP socket and the two 3.3K ohm resistors to the PCB. Next, solder one end of the 16-pin DIP jumper to the PCB, being careful to observe the polarity shown in Fig. 4. (Pin 1 of the DIP jumper is identified by a mark.) Now, attach each of the 19 leads from the 40-pin cable to the PCB as shown in Fig. 4, being careful to ensure that the proper lead goes in the proper hole.

Get an 18-inch length of thin coaxial (audio or microphone) cable. Strip 1/2-inch of the outer jacket from each end. Then strip 1/8-inch of insulation from the two leads on each end. Place the center conductor of one end into the PCB hole marked "+" and solder. Place the braid from the same end into the PCB hole marked "–" and solder. Place the shell of the power connector over the free end. Solder the center conductor of the free end to the short (center) lug of the power connector. Solder the braid to the remaining power connector lug and reattach the shell. Holding the 6821 by its ends, insert it into the 40-pin socket, observing the orientation shown in Fig. 4. The PCB is now ready to mount in the case.

Before mounting the PCB, final preparation of the case is required. Insert the four 4-40 by 3/4-inch machine screws into the four mounting holes in the case so the screw heads are on the underside of the case. Secure each screw with a machine nut. Now place another machine nut on each screw about 1/2-inch up from the bottom of the case.

Using a small round file, make a semicircular depression on the case's front top, 3/4-inch from the right edge. Make it deep enough to allow the power cable to exit the case. Mark the front of the case as you wish.

Place the PCB in the case, foil side up, mating the PCB's mounting holes with the four 3/4-inch screws. The PCB will now rest on the four machine nuts. Place four more machine nuts on the screws to secure the PCB. Bring the 40-pin cable out the back of the case, and the 16-pin DIP and power cables out the front of the case (the power ca-

Answer: Smith-Corona

Question: What company offers a new daisy wheel printer, three dot matrix printers and a combination printer-typewriter, with suggested retail pricing of $395 to $795?

Question: What printer company offers print quality that challenges printers costing hundreds of dollars more?

Question: What printer company offers dual interfaces for all five of its printer models?

Question: What printer company offers removable and adjustable tractor feeds as standard equipment on all of its dot matrix models?

Question: What printer company has a toll-free telephone number to call if you ever have a problem? And an extensive service system, too?

D-300 (TM) dot matrix printer.

L-1000 (TM) daisy wheel printer.

Ultrasonic III Messenger (TM) portable typewriter with optional Messenger Module.

D-100 (TM) dot matrix printer.

D-200 (TM) dot matrix printer.

☐ Please send me more information about Smith-Corona printers; I am interested in in-home use.

☐ Please send me more information about Smith-Corona printers for office use.

Name _____

Company Name _____

Business Address _____

City _____ State _____ Zip _____

Type of Business _____

Send to: Jerry Diener, V.P. Sales, Smith-Corona
65 Locust Avenue
New Canaan, Connecticut 06840

SMITH-CORONA® HC-7

ble should rest in the semicircular depression in the case front, and be to the right of the 16-pin DIP cable). Secure the top and bottom case halves together with the hardware provided. This completes construction of the master interface.

Testing the Master Interface

Before you begin testing, there is one important rule you should always follow to avoid damaging your CoCo. Never connect or disconnect the master interface when power is applied to the CoCo. To connect, first make sure power is off. Then, lift up the cartridge slot door and insert the 40-pin connector so pin 1 is up and to the rear of the CoCo.

Carefully insert the 40-pin connector into the cartridge connector. If you do this properly, you will feel a positive engagement. To remove the 40-pin cable, first make sure power is off. Then, open the cartridge slot door and grasp the connector. Pull out the connector.

You will need a logic probe (or alternate) to check the status of the DIP pins. If you have one, you can power it from the lugs of the power connector. If you don't have a logic probe, you can make a simple tester from a 470-ohm resistor, a light-emitting diode (LED), and a jumper wire. (See Table 3.)

All LEDs have an anode and a cathode. To make the LED light, you must apply a positive voltage to the anode. Determine which lead is the anode and which is the cathode. Connect the cathode to one end of the 470-ohm resistor. Connect the free end of the 470-ohm resistor to the ground lug of the power connector (you can use a jumper wire for temporary connection). The free (anode) end of the LED will serve as your probe.

Type in and CSAVE Program Listing 1. Remove power from your CoCo, attach the Master Interface, and repower the CoCo. Using a voltmeter, check between the two lugs of the power connector for the presence of 5 volts. (The actual value can be as low as 4.75 volts.) Do not continue until you verify this.

Then, CLOAD and run the program. A representation of the 16-pin DIP cable will appear on the screen. You see the DIP header with the pins facing you, and the cable exiting to the right. The pin numbers are enclosed in square brackets. Between each pin number and pin is either a zero or a one. A zero indicates that the pin is low (zero volts); a one indicates that the pin is high (5 volts).

Near the top of the screen are two legends; PA equals zero and PB equals zero. PA and PB are the decimal numbers that represent the state of the peripheral data lines on sides A and B of the PIA. The program initially configures all 16 lines as output and sets them to logic zero. At the bottom of the screen is a prompt stating "Toggle which pin (1-16)?"

When you enter a pin number, that pin's status will change state (low to high or high to low). Enter one and note that the status of pin 1 changes from zero to a one. Also note that PB now equals 128. This is because pin 1 is bit 7 of side B.

Now, place the tip of your logic probe (or the anode of the LED) on pin 1. Note that it glows, indicating that pin 1 is high. Try all the other pins; they should all indicate low (i.e., no light). Toggle each of the pins in turn, checking for proper status with your tester.

When you toggle a pin, the program POKEs the new status to the interface. It then PEEKs both sides A and B and compares them to what they should be. If the PEEKs and POKEs do not match, the program will flash ERROR on the screen. This can indicate that there is a typing error in the program, or a fault of some kind in the interface. In either instance, you will have to troubleshoot to determine what is wrong.

The most likely candidate is a typing error. Check the program for errors and correct any you find. Try the program again. If you still get an error, check your interface for cold solder joints, unsoldered connections, solder bridges, and the like. Correct any faults you find and try again.

The Program

When everything works properly, you have a fully functioning interface. Now you can relax and take a closer look at the program to see how it works. This program approach is neither the best nor the only way to communicate with the interface. Actually, lines 30, 40, 230, and 240 are the only lines that communicate with the interface; the remainder perform screen and data formatting.

Let's take a line-by-line look, since it will help you understand the interface workings better and should give you some ideas of your own.

Line 10 defines three 16-element arrays. These are B (binary value), P (screen position) and V (value). Line 20 prints a title. Line 30 initializes side A of the interface's PIA. First, it POKEs a zero to $C001 to select the data-direction register. Next, it POKEs 255 to $C000 to configure all side A lines as outputs. With the peripheral lines configured, it POKEs a four to $C001 (setting bit 2 of the control register) to select the peripheral register. Then, it POKEs a zero to $C000 to set all lines low.

Reference	Description	RS Part #
R1, R2	3.3K ohm, ¼-watt fixed resistor	271-1328
SO1	40-pin IC socket	276-1996
W1	16-pin DIP jumper cable	276-1976
P1	Coaxial DC power jack (5mm OD, 2.1mm ID)	274-1565
-	Aluminum case	270-271
-	4-40-by-¾" machine screw (8 req'd)	64-3011
-	4-40 machine hex nut (12 req'd)	64-3018
IC1	6821 PIA (Spectrum Projects Part #6821)*	
W2	40-pin cable (Spectrum Projects Part #40W2)*	

*Available from Spectrum Projects, 93-15 86th Drive, Woodhaven, NY 11421, for $14.95 plus $3 shipping and handling. NY state residents add appropriate sales tax.

Table 2. Master Interface List of Materials

Reference	Description	RS Part #
LD1	T-1 or T-1¾ red light-emitting diode	276-041
RL	470-ohm, ¼-watt fixed resistor	271-1316

Miscellaneous: Printed circuit board (see text), 18 inches of ⅛-inch to ¼-inch diameter audio-type coax cable, solder, etc.

Table 3. LED Tester Parts

Line 40 performs the identical function for side B. Lines 50 and 60 fill the P array with screen positions. Line 70 fills the B array with the decimal value associated with the DIP pin. For instance, B(8) is set to one (two raised to the zero power), since pin 8 is bit 0 of side B. This array will let you quickly calculate the decimal number to be POKEd when you want to change the status of the peripheral lines. Lines 80–160 format the screen.

Line 170 retrieves your pin selection, while line 180 prohibits inputs less than one or greater than 16. Line 190 changes the value stored in the V array for that pin. For instance, say pin 4 was a zero (V(4) = 0). Line 190 would calculate the truth test (V(4) = 0). Since this is true, a −1 is returned. Then, −1 × −1 = +1. If V(4) were a one, the truth test would return a zero. Then 0 × −1 = 0. So in either instance, the value is toggled between zero and one. Line 190 also reinitializes variables PA and PB.

Line 200 calculates the decimal number to be POKEd to $C000 based on the pin status you request. It multiplies the value of the pin by the decimal number associated with that pin and keeps a running total in PB. For instance, if pins 2 and 5 were high (one) and all others were low (zero), PB would equal the sum of 1 × 64 (since B(2) = 64) and 1 × 8 (since B(5) = 8). This value of 72 would be shown on the screen, and used in line 230 to change the status of the peripheral lines. Line 210 performs the same function for PA. Then line 220 prints the results to the screen.

Line 240 POKEs the new values of PA and PB to the interface. Line 250 then PEEKs the interface to check that the interface lines were correctly set to these values. If they were, program execution returns to line 150 where the screen is updated and a new input request provided. If the PEEKed information does not match PA and PB, execution continues to the endless loop created by lines 250 and 260. Here an error message flashes on the screen, notifying you that something is wrong.

"I'll also be building a functional robot arm."

Next Time

Notice that the test program did not exercise the interface as an input device, and I didn't provide any useful application for it. These two areas will both be corrected next time when you begin your first project. It is a selfish one; something that I always wanted. It's an IC tester that can check most ICs with 16 or fewer pins. To get you started, I'm including a list of parts (Table 4) so you can begin as soon as possible. I'll continue this procedure, since future projects will require mail-order parts purchase. By ordering the parts ahead of time, you'll have them in time to start building immediately.

The Future

I'll show you how to build a small "mouse," one of those two-wheeled, motorized devices with programmable movement (sort of Logo brought to life). I'll also be building a functional robot arm. Both these projects will require some simple woodworking skills (like using a saw and hammer).

The nature of these animation projects tends to make them a bit more costly, so I'd like to hear from you. Any suggestions or comments on any part of this series are welcomed. I'll answer each letter as long as you provide a self-addressed, stamped envelope. ■

Address correspondence to James J. Barbarello, R.D. #1, Box 241 H, Tennent Road, Englishtown, NJ 07726.

Reference	Description	RS Part #
R1–R16	2.2K ohm, ¼-watt resistor (TTL version)	271-1325
	10K ohm, ¼-watt resistor (CMOS version)	271-1335
DS1, DS2	8-position, SPST DIP switch	275-1301
TJ1–TJ16	Printed circuit board test jack	274-728
PT1, PT2	Solderless phone tip	274-723
SO1, SO2	16-pin DIP socket	276-1998
J1	Chassis mount jack (5mm OD, 2.1mm ID)	274-1549

Miscellaneous: Printed circuit board, stranded hookup wire, solder, case, etc.

Table 4. IC Tester Project List of Materials

↙ See List of Advertisers on page 99

BY MICHAEL L. JOHNSON

VARIABLE CROSS-REFERENCER

Knowing which variables your software contains is the first step toward streamlining a program.

As a relative newcomer to the Color Computer, I first set out to learn more about the inner workings of Extended Color Basic. After PEEKing around in memory for a while, I hit upon the locations where Basic programs are stored.

Using Program Listing 1, I determined the meaning of each byte and with more PEEKing and POKEing, I derived Fig. 1.

Each Basic line consists of the following:

- starting address of next line
- line number
- tokenized program line
- null character to signify the end of the line.

I began to explore applications of this knowledge. The easiest and most useful application was the variable cross-referencing program in Listing 2. This program finds and cross-references variables in other programs.

```
100 CLS
110 X=5121
120 :  X1=PEEK(X)
130 :  H$=HEX$(X1)
140 :  X$=CHR$(X1)
150 PRINT HEX$(X);TAB(7);X1;TAB(
14);X$;TAB(21);H$
160 X=X+1
170 GOTO120
```

Program Listing 1

Program Description

The first part of Listing 2 initializes the memory and variables. A PCLEAR 1 statement ensures that the program is located at the same point in memory each time. CLEAR 2000 and DIM CR$(30) allow space for the cross-reference table. If the program you are examining has more than 30 variables, be sure to change line 60030 accordingly.

Line 60150 loads LO with 5121, which is the start of program memory in 16K Disk Basic. For Extended Color Basic, this number should be 3073.

Line 60260 loads LI with the current line number. It stores the line number in 2-byte format. The following formula extracts this number:

high byte × 256 + low byte.

Lines 60300–60330 load A$ with the contents of the current line. Line 60300 checks to see if the line is a remark and skips it if it is.

One problem with searching for variables is that the Color Computer does not distinguish between characters in a string and characters that designate a variable. To avoid problems, lines 60350–60410 delete the literal strings. Quotation marks set off the literal strings, so that when the pro-

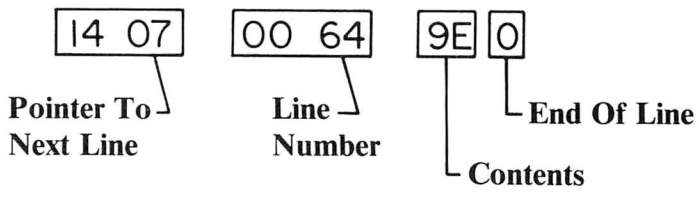


Fig. 1. Structure of a Basic Line

System Requirements

16K RAM
Extended Color Basic

gram reads a quotation mark in this section, it deletes characters until it reaches a second quotation mark. The program, therefore, deletes the characters in the print statements, but leaves the variables intact. Since the program is tokenized, only the variables and numbers are left in ASCII format.

To distinguish between variables and other characters, line 60450 determines whether the first character is the first character of a variable. If it is, the program jumps to the routine at 60630. Otherwise, it continues its search for variables.

At line 60630 the search becomes a bit more complex. If the next character is a number or a letter, the program adds it to the name of the variable and loops back for more. If the character is a $ or a space, the program adds it to the variable and then looks for an open parenthesis to signify an array. When the program finds any other character, it prints the variable and adds it to the list.

"...the most important use of the cross-reference table is in streamlining other programs."

If the program finds an open parenthesis, it adds the characters inside and including the parentheses to the variable. From there, the routine starting at line 60510 changes plus signs, minus signs, slash marks, and asterisks from the tokenized form that represents them to the characters that you can recognize. The program does not list variables that are inside the parentheses separately.

Line 60570 prints the variable on the screen and keeps the screen format neat. It then jumps to 61070, which adds the variable and its new location to the cross-reference list.

The loop from 61070–61100 determines whether the variable has been used yet and if the program already has a listing for the current line number. This prevents multiple-line listings in the cross-reference table.

If the program does not have a listing for the current line number, it adds the line number to the line specifying that variable. If the variable hasn't been used yet, the program adds it, right-justified, to the end of the cross-reference table. The program returns to line 60600, where it clears VA$, then it continues the search for another variable.

Once the program finishes one line, it loads the next line pointer (NE) into LO, clears A$, and searches another line. After it has searched all the lines, the program sorts the CR$ array, which holds the contents of the cross-

```
60000 CLS
60010 CLEAR 2000
60020 PCLEAR 1
60030 DIM CR$(300)
60040 PRINT @228,"VARIABLE CROSS
 REFERENCE"
60050 PRINT @264,"BY MIKE JOHNSO
N"
60060 PRINT @300,"02/20/83"
60070 PRINT @329,"FOR TRS-80 COL
OR                    16K"
60080 FOR X=1 TO 2000
60090 IF INKEY$="" THEN NEXT X
60100 REM VARIABLE CROSS REFEREN
CE
60110 REM BY MIKE JOHNSON
60120 REM 2/20/83
60130 REM
60140 CLS
60150 LO=5121
60160 REM LO=5121 FOR CO.CO.
          WITH 16K DISK BASIC
          MAY BE DIFFERENT WITH
          OTHER CONFIGURATIONS
60170 NE=(PEEK(LO)*256+PEEK(LO+1
))
60180 IF NE=0 THEN 60780
60190 C=C+1:IF C<>10 THEN 60250
60200 PRINT:PRINT
60210 PRINT"PRESS ENTER TO CONTI
NUE"
60220 IF INKEY$<>CHR$(13) THEN
      60220
60230 C=0
60240 CLS
60250 LO=LO+2
60260 LI=(PEEK(LO)*256+PEEK(LO+1
))
60270 IF LI=60000 THEN 60770
60280 PRINT STRING$(6-LEN(STR$(L
I))," ");LI;
60290 REM LOAD STRING WITH LINE
60300 IF PEEK(LO+2)=130 THEN 606
20
60310 FOR I=LO+2 TO NE-1
60320 :   A$=A$+CHR$(PEEK(I))
60330 NEXT I
60340 REM STRIP THE STRING OF
          LITERAL STRINGS
60350 I=1
60360 IF MID$(A$,I,1)=CHR$(34)
      THEN 60390
60370 I=I+1:IFI>=LEN(A$)THEN 604
30
60380 GOTO 60360
60390 I=I+1
60400 IF MID$(A$,I,1)=CHR$(34)
      THEN 60370
60410 A$=LEFT$(A$,I-1)+MID$(A$,I
+1):GOTO 60400
60420 REM FIND LEGAL VARIABLES
60430 I=1
60440 CH$=MID$(A$,I,1):CH=ASC(CH
$)
60450 IF CH>64 AND CH<90 THEN 60
630
60460 IF CH=0 THEN 60620
60470 I=I+1
60480 IF I=>LEN(A$) THEN 60620
60490 GOTO 60440
60500 FOR J=1 TO LEN(VA$)
60510 :   IF MID$(VA$,J,1)<="Z"TH
EN          60560
60520 :   IFMID$(VA$,J,1)=CHR$(17
2)          THEN MID$(VA$,J,1)="-
"
60530 :   IFMID$(VA$,J,1)=CHR$(17
1)          THEN MID$(VA$,J,1)="+
"
60540 :   IFMID$(VA$,J,1)=CHR$(17
4)          THEN MID$(VA$,J,1)="/
"
60550 :   IFMID$(VA$,J,1)=CHR$(17
3)          THEN MID$(VA$,J,1)="*
"
60560 NEXT J
60570 PRINT TAB(6*C2+8);VA$;
60580 C2=C2+1:IF C2<>4 THEN
      60590 : ELSE C2=0 : PRINT
60590 GOTO 61070
60600 VA$="":GOTO 60480
60610 REM CLEAR STRING, LOAD NEW
          POINTER, START OVER
60620 C2=0:LO=NE:PRINT:A$="":GOT
O      60170
60630 VA$=CH$
60640 I=I+1
60650 CH$=MID$(A$,I,1)
60660 CH=ASC(CH$)
60670 IF CH=36 OR (CH>64 AND CH<
91) OR (CH>47 AND CH<58) THEN 60
750
60680 IF CH=0 THEN 60570
60690 IF CH<>40 AND CH<>32
      THEN 60500
60700 IF CH=32 THEN 60640
60710 VA$=VA$+MID$(A$,I,1)
60720 IF MID$(A$,I,1)=")" THEN 6
0500
60730 I=I+1
60740 GOTO 60710
60750 VA$=VA$+CH$
60760 GOTO 60640
60770 REM SORT VARIABLES
60780 FOR B=1 TO N-1
60790 :   F=0
60800 :   FOR C=1 TO N-1
60810 :       IFCR$(C+1)>=CR$(C)TH
EN             60860
60820 :         T$=CR$(C)
60830 :         CR$(C)=CR$(C+1)
60840 :         CR$(C+1)=T$
60850 :         F=1
60860 :   NEXT C
60870 :   IF F=0 THEN 60890
60880 NEXT B
60890 PRINT"PRESS ENTER FOR CROS
S REFERENCE"
60900 IF INKEY$<>CHR$(13) THEN 6
0900
60905 CLS
60910 FOR I=1 TO N+1
60920 :   PRINT CR$(I)
60930 :   C3=C3+LEN(CR$(I))
60940 :   IF C3=>256 THEN 60970
60950 NEXT I
60960 END
60970 PRINT
60980 C3=0
61030 PRINT"PRESS ENTER TO CONTI
NUE"
61040 IF INKEY$<>CHR$(13) THEN 6
1040
61050 CLS
61060 GOTO 60940
61070 FOR J=1 TO N
61080 IF STRING$(8-LEN(VA$),32)+
VA$=LEFT$(CR$(J),8) THEN FL=1:IF
 STR$(LI)<>RIGHT$(CR$(J),LEN(STR
$(LI))) THEN CR$(J)=CR$(J)+STRIN
G$(8-LEN(STR$(LI)),32)+STR$(LI):
GOTO 61130
61090 IF FL=1 THEN 61130
61100 NEXT J
61110 N=N+1
61120 CR$(N)=STRING$(8-LEN(VA$),
32)+VA$+STRING$(6-LEN(STR$(LI)),
32)+STR$(LI)
61130 FL=0:GOTO 60600
```

Program Listing 2. Cross-Referencing Program

reference table, first by length and then alphabetically. Finally, the program prints the list of variables.

Other Considerations

A few errors might occur with the cross-referencing program. First, the BS error indicates that the program you are examining has too many variables. Change line 60030 accordingly. Second, if a variable appears too often, one of the variables in the cross-reference table will exceed the allowed length of 255 characters.

In addition, if you own a disk drive, you can merge this program with the program you want to cross reference, so you do not have to enter the program each time you want to use it.

Perhaps the most important use of the cross-reference table is in streamlining other programs. Because it determines which variables appear in each line, it lets you delete variables that are used for the same purpose, thus conserving precious string space. ■

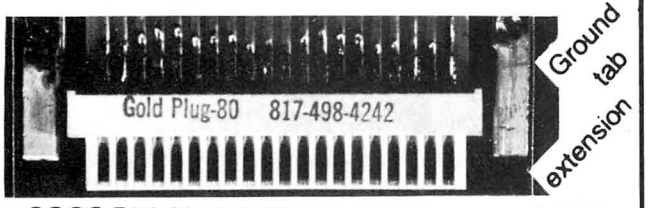
Address correspondence to Michael Johnson, 7481 Greenway Drive, Jacksonville, FL 32244.

↙ See List of Advertisers on page 99

TRS-80+ MOD I, III, COCO, TI99/4a
TIMEX 1000, OSBORNE, others

GOLD PLUG - 80

Eliminate disk reboots and data loss due to oxidized contacts at the card edge connectors.
GOLD PLUG 80 solders to the board edge connector. Use your existing cables. (if gold plated)



COCO Disk Module (2)	new	$16.95
Ground tab extensions		INCL
Disk Drives (all R.S.)	SPECIAL PRICE	$7.95
Gold Disk Cable 2 Drive		29.95
Four Drive Cable		39.95

USA shipping $1.45 Can/Mex $4.
Foreign $7. **Don't wait any longer** TEXAS 5% TAX

Available at your favorite dealer or order direct from

MasterCard **E.A.P. CO.** VISA
P.O. BOX 14
KELLER, TEXAS 76248
(817) 498-4242 MC/VISA
+ trademark Tandy Corp ↙216

STORM TRAX

***Hurricane Analysis System**

Gulf * Atlantic
* Caribbean *

32 K Color Computer, Ext. Basic
Cassette or Disk

Menu Driven
All Graphics Mode
High Resolution Maps

Loads and Stores DATA FILES
Produces LISTINGS
PLOTS Storm Track, Winds and Pressure
COMPUTES TRUE Heading, Speed, and Distance Data
plus *THREAT FACTORS for 10 Cities*

TAPE $19.95 DISK $24.95
1978-83 Storm Data; Tape 3.95, Disk 6.95

Send SASE for More Information ↙426

LOGIX SYSTEMS Lmtd.
P.O. BOX 57012 WEBSTER, TEXAS — 77598

KEY—264K

- ACCESS YOUR 64K RAM AS TWO 32K BANKS FROM BASIC IT'S LIKE HAVING TWO COMPUTERS IN ONE !!!
- HAVE SEPARATE PROGRAMS IN EACH BANK AND SWITCH INSTANTLY BETWEEN THEM WITH SIMPLE KEYSTROKES
- HAVE ONE LARGE PROGRAM THAT OCCUPIES BOTH BANKS
- RUN TWO PROGRAMS AT THE SAME TIME WITH FOREGROUND/BACKGROUND MULTI-TASKING
- ADDS 16 NEW COMMANDS TO EXTENDED OR DISK BASIC PASS VARIABLES BETWEEN BANKS - CALL SUBROUTINES ACROSS BANKS - VIEW TEXT OR GRAPHICS FROM EITHER BANK - COPY MEMORY ACROSS BANKS, START AND STOP MULTI-TASKING, ALL WITH BASIC COMMANDS
- INCLUDES 8 KEYBOARD COMMANDS TO ALLOW SWITCHING BANKS, MULTI-TASKING, BREAK, RESET, COLD STARTS AND DUPLICATING ONE BANK TO THE OTHER
- WORKS WITH CASSETTE OR DISK BASED SYSTEMS
- WORKS ON ANY 32K OR 64K COCO WITH EXTENDED OR DISK BASIC AND GOOD 64K MEMORY CHIPS

ORDER YOUR KEY-264K TODAY by sending check or money order for $39.95 (Cassette) or $44.95 (Disk) plus $2.00 postage U.S.A. ($5.00 outside U.S.A.) Mass. residents add 5% sales tax. COD (add $3.00), MASTERCARD, or VISA call (617) 263-1737

KEY COLOR SOFTWARE
P.O. BOX 360
HARVARD, MA. 01451 ↙48

WANTED!

Young men and women seeking adventure, excitement and thrill-a-minute action. No experience necessary—just you and your Color Computer. See below:

FOR THE 32K THRILLSEEKER

	TITLE	DISK	CASS.
NEW!	MR. DIG	$27.95	$25.95
NEW!	CANDY CO.	29.95	29.95
NEW!	WILLIE'S WAREHOUSE	29.95	29.95
NEW!	ICE HOCKEY	——	21.95
	ZAXXON	29.95	——
	PROTECTOR II	——	29.95
	DESERT PATROL	——	21.95
	ICEMASTER	——	21.95
	FOODWAR	——	22.95
	WACKY FOOD	——	19.95
	CASHMAN	——	24.95
	CHOPPER STRIKE	——	24.95
	TIME BANDIT	——	24.95

LOTS OF PLAY FOR 16K

MOONSHUTTLE	——	$26.95
FROG TREK	16.95	14.95
3-D TIC-TAC-TOE	18.95	16.95

Interested applicants send check or m/o to:

OELRICH PUBLICATIONS, INC.
4040 N. NASHVILLE
CHICAGO, IL 60634

Credit card orders call: 800-621-0105
(In Illinois call: 312-545-9286)
NO SHIPPING CHARGES!!! ↙288

HOT PROGRAMS IN AN INSTANT

Finally! Great programs you can load and run in seconds. You'll save hours of keyboarding time and build a terrific library of useful programs for as little as $10.00 a month.

Get the best programs from HOT CoCo magazine... ready to run...for pennies. All the directions or help you need is in your copy of HOT CoCo.

instant CoCo gives you ...*action-packed games—have your own video game entertainment center!

*useful applications—programs for education, business, home and hobby, to delight the whole family.

*utilities—your routine tasks will be taken care of like magic plus you'll be expanding the capabilities of your computer.

Order **instant CoCo** cassettes by the month. Every cassette guarantees you ten to fifteen great programs for only $11.47.*

Save money! A full twelve month subscription is yours for only $99.97*. That's only $8.33 per cassette.

You'll want the "Best of 83."

Over 33 programs from the 1983 issues of HOT CoCo magazine. You'll have a valuable library...for only-$16.47*...less than .45 cents per program.

Order your subscription to **instant CoCo** today!

Mail the coupon below or call TOLL FREE 1-800-258-5473 and use your Visa, Mastercard or American Express. In New Hampshire call 1-924-9471.

instant CoCo lets you start having fun right away...almost like magic!

YES! I want programs in an instant... I want instant CoCo

☐Please send me____copies of the "Best of '83" at $16.47 each.

☐Please send me ____copies of this month's issue at $11.47 each.

☐Please sign me up for a one year subscription beginning with this month's issue at $99.97

*Price includes postage and handling. Foreign Air Mail please add an additional 45¢ per cassette or $25 per subscription. US funds drawn on US banks ONLY.

☐CHECK/MO ☐MC ☐VISA ☐AE

Card# _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

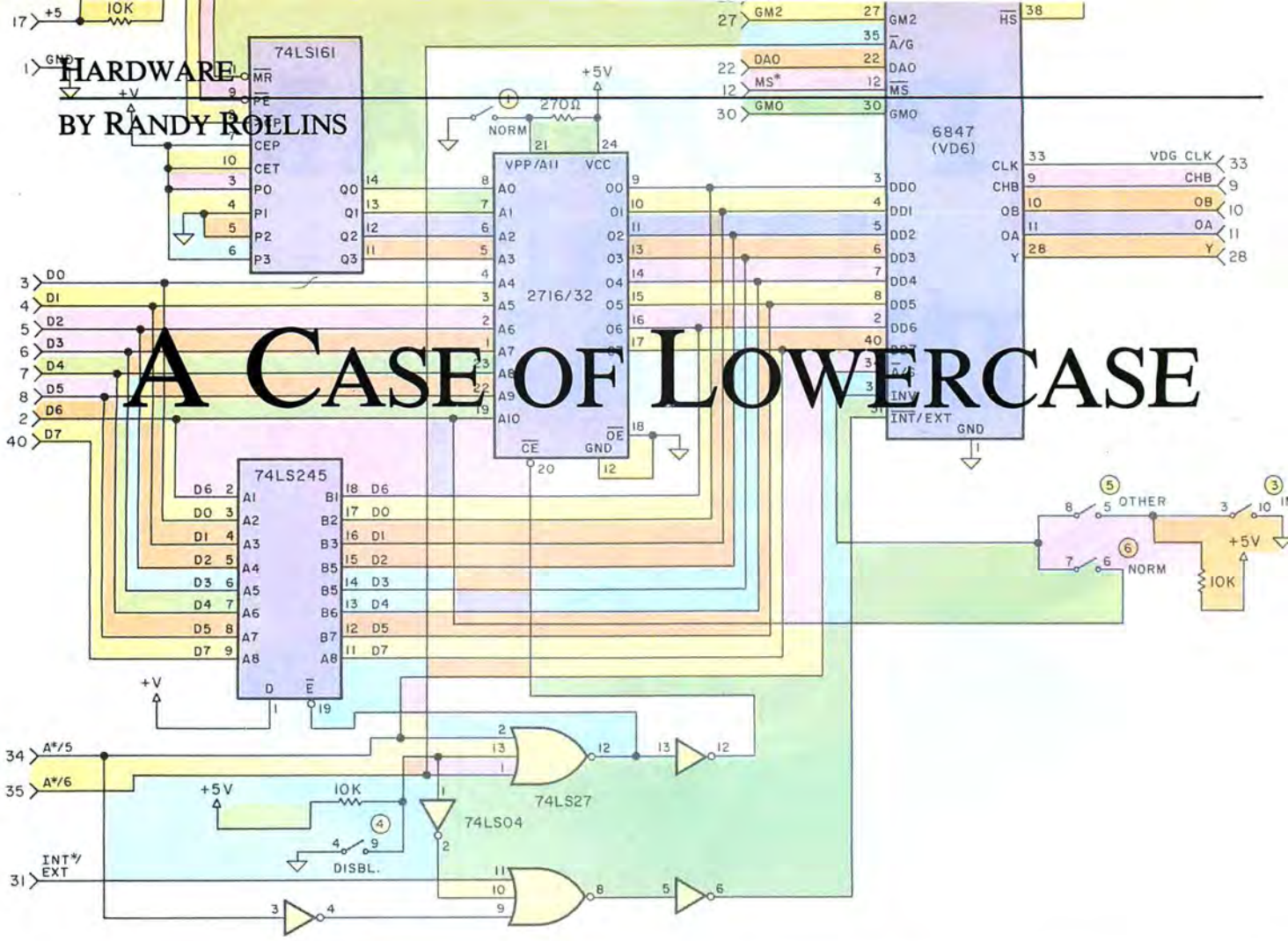
Please allow 6–8 weeks delivery.

instant CoCo • 80 Pine Street • Peterborough, NH 03458

IC8407

BY RANDY ROLLINS

A CASE OF LOWERCASE



When you want to make the Color Computer an even better machine, what can you do? A lowercase letter generator in hardware is an important refinement, and CoCo hardware adapts easily to its addition. This article helps you make such a refinement and explores how the Video Display Generator works in generating text to screen.

There are several ways to create lowercase letters using the CoCo graphics modes, but these methods rely on the use of software, and use too much memory. Often they are slow in through-put, and hard to read if they generate more than 45 letters across the screen. A lowercase generator in hardware is a practical solution.

The Color Computer video display generator (VDG) chip, the 6847 (see Fig. 1 for pin-outs), is a complex device containing the internal circuitry needed to display a video signal, in color, as alphanumeric or semi-graphics characters, or full bit-mapped graphics.

The alphanumeric display mode has both internal and external modes. Normally, the CoCo uses the internal mode,

You can improve on a good thing. Build this lowercase mod and get an attractive character set.

which lets the VDG use its internal alpha-generator ROM containing the 64 alpha characters.

To understand how the internal or external alpha generator works, you need to know how the data is presented on screen. In the VDG's normal text mode, you display a full hi-res data screen because each of the 32 characters displayed in a row is made up of a 5-by-7-pixel field within an 8-by-12-pixel field (Fig. 2).

There are eight horizontal pixels of resolution in each character ($8 \times 32 = 256$ pixels horizontally). There are 12 vertical pixels in each character ($12 \times 16 = 192$). Therefore, there are 256 horizontal by 192 vertical pixels, or the highest screen resolution when displaying 32 characters by 16 rows.

To display alpha data the computer scans a ROM addressed by the screen data in RAM to create characters (Fig. 3). This ROM can either be the internal VDG ROM, or an external one.

The internal VDG character ROM stores 64 character patterns, each with a specific address (Table 1). When the Color Computer displays text, it scans RAM addresses 1024 ($400) through 1535 ($5FF). The data in these locations forms the address of the character stored in ROM.

You can see how this works if you run the listing below:

```
10 CLS
20 PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT:PRINT
30 REM ROUTINE TO DUMP CHARAC-
   TERS IN VDG ROM TO SCREEN
40 FOR I=0 TO 127
50 POKE I+1024,I
60 NEXT I
70 END
```

The program displays two character sets: the normal dark characters on a light background, and the inverted light characters on a dark background. The inverted characters are the 64 stored in the VDG. The invert line (INV), pin 32,

is set high (Fig. 1) because when bit 6 is high it pulls up this line. Since the VDG ROM stores only 64 characters, the computer only displays 128 characters. Though this is a clever way to double the effective number of characters, it doesn't work for displaying lowercase.

Look now at what you can do with an external ROM generator. Select the external generator by putting pin 31 of the VDG chip high, which causes the VDG to look for data on its input pins rather than in the internal generator ROM.

The computer then latches and displays this data in the same 8-by-12 format mentioned earlier (32 characters by 16 rows). The data gets to the VDG by an external generator ROM, or in this case, an EPROM and circuitry that handles the interface of the external generator. Figure 4 is a full schematic.

Circuit Description

The heart of the circuit is the VDG, and next in importance is the EPROM that holds the new set of characters. (Table 2 lists its contents.)

The alpha display memory data ($400–$5FF) and the output of the 74LS161 counter address the EPROM, and the counter counts the character cell's 12 rows. For example, your TV display is made of small horizontal lines (196 for the CoCo's display). Each time the computer scans a line it generates a horizontal sync pulse that increments the counter by one. Since each character cell contains 12 rows, the VDG sends out a clear signal setting the counter back to zero after 12 counts.

This process repeats for all 16 rows of characters (196 counts). At the end of the screen the computer generates a field sync (FS) that presets the counter to nine, keeping the top line together with the rest of the screen. Thus data coming from display memory, and the counter read out the EPROM data.

Now that you know how to use an external character generator for the VDG chip there are details concerning my circuit that you must understand. When the VDG is used in the graphics and semigraphics modes, it explicitly interprets the data presented at the data inputs. Therefore, you must disconnect the external character generator from the VDG input and connect the data from the display memory, or graphics memory. Do this with a tri-state bus buffer (74LS245) and the associated logic (74LS27 and 74LS04). Next, since you need to enable or disable the external generator circuit without a lot of unplugging, the logic for the tri-state enable is coupled to switch S1 (see Fig. 4). You can also select the normal alpha display mode from the standard (lowercase inverted), or choose all characters inverted or all not inverted.

The all inverted characters are the most pleasing to look at, since with a good brightness and contrast adjust-

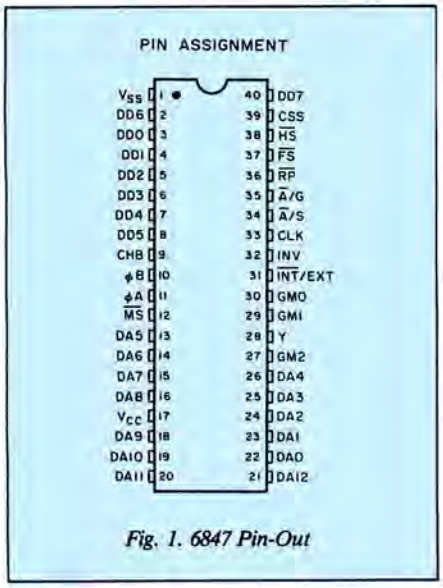


Fig. 1. 6847 Pin-Out

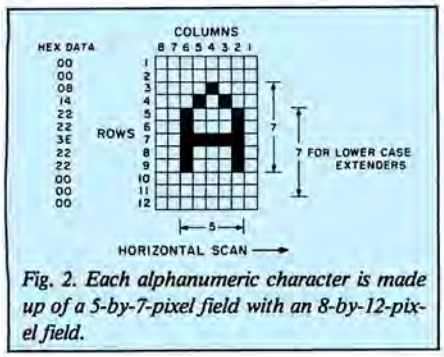


Fig. 2. Each alphanumeric character is made up of a 5-by-7-pixel field with an 8-by-12-pixel field.

ADDRESS
Upper 4 bits: Lower 4 bits

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
CoCo	0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
inverted	1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]		
characters	2	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
	3	0	1	2	3	4	5	6	7	8	9	:	;	<	>	=	?
CoCo	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
normal	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]		
characters	6	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
	7	0	1	2	3	4	5	6	7	8	9	:	;	<	>	=	?

Table 1. Internal VDG Character Set

Qty.	Description	Approx. Price
1	74LS161 binary counter	.69
1	74LS245 bus transceiver	$1.49
1	74LS27 triple 3-in. NOR	.29
1	74LS04 hex inverter	.35
1	2716 or 2732 350ns access time or less	$5.95 (2716)/$7.95 (2732)
1	7-switch DIP switch	$1.39
3	10k-ohm 1/4-w resistors	.30
1	270-ohm 1/4-w resistor	.30
3	40-pin sockets	$1.00 ea
1	20-pin socket	.45
1	16-pin socket	.35
3	14-pin sockets	.29 ea
6	.01-μF capacitors	.08 ea
1	2.75 × 3.75 grid board (Radio Shack 276-158)	$1.95
1	50-foot spool of 30 gauge wire	$2.39
1	40-pin DIP socket adapter. Samtec P/N APO-640-TH or Augat P/N 640-AG3T	$3.50
		Approx. Total $17.75 (2716)

Prices are taken from the 1984 Jameco and Radio Shack catalogs. The socket adapter is not available from Jameco or Radio Shack, but most good electronic supply stores carry them.

Table 2. Parts List for Lowercase Generator Board

ment the display looks like a green phosphor monitor. With the circuit shown using a 2732 EPROM you can store two switch-selectable character sets.

Construction

You do not need to modify your computer's printed circuit card to install this upgrade. The circuit is built on a small board that plugs into the VDG socket (Fig. 5). The VDG, in turn, plugs into a socket on the small board.

This type of installation requires that you remove the top of the RF cage in the older CoCos. You won't have this problem with the TDP-100s and F revision of the CoCo's PC card since the RF cage is only around the RAM and SAM chips.

When I built this circuit I used tight point-to-point wiring, which works if you use a clean wiring technique. Your wires should be as short as possible, and be careful how you route the clock lines. It is also important to use sufficient decoupling. It is adequate to place a .01μF capacitor on each chip as close to the +5 and GND as possible. If you are not fond of wiring things I can lay out a PC card for you. I would be willing to send a burned EPROM (a 2716) with the data shown in Table 2 for $12. Note that the EPROM should have an access time of 350ns, or the characters will not be clear. ■

Address correspondence to Randy Rollins, 6119 Brookhollow Drive, West Valley, UT 84120.

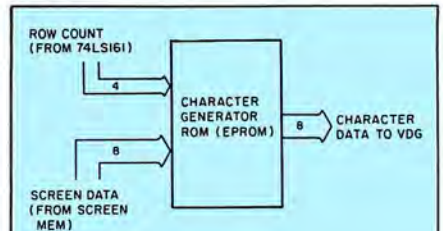


Fig. 3. To display alpha data the computer scans a ROM addressed by the screen data in RAM to create characters.

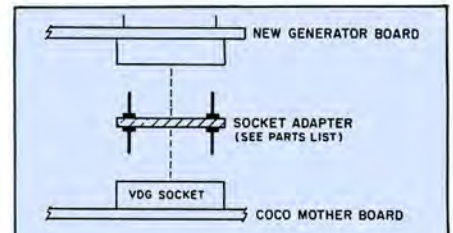


Fig. 5. Board Installation Detail

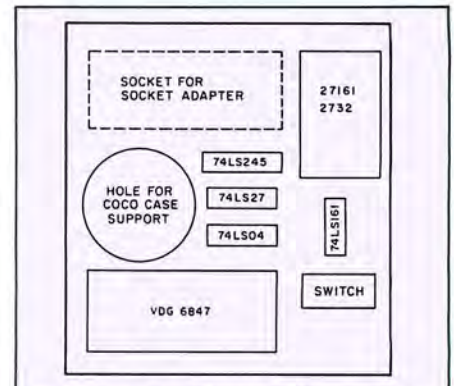


Fig. 6. Basic Board Layout (old version CoCo)

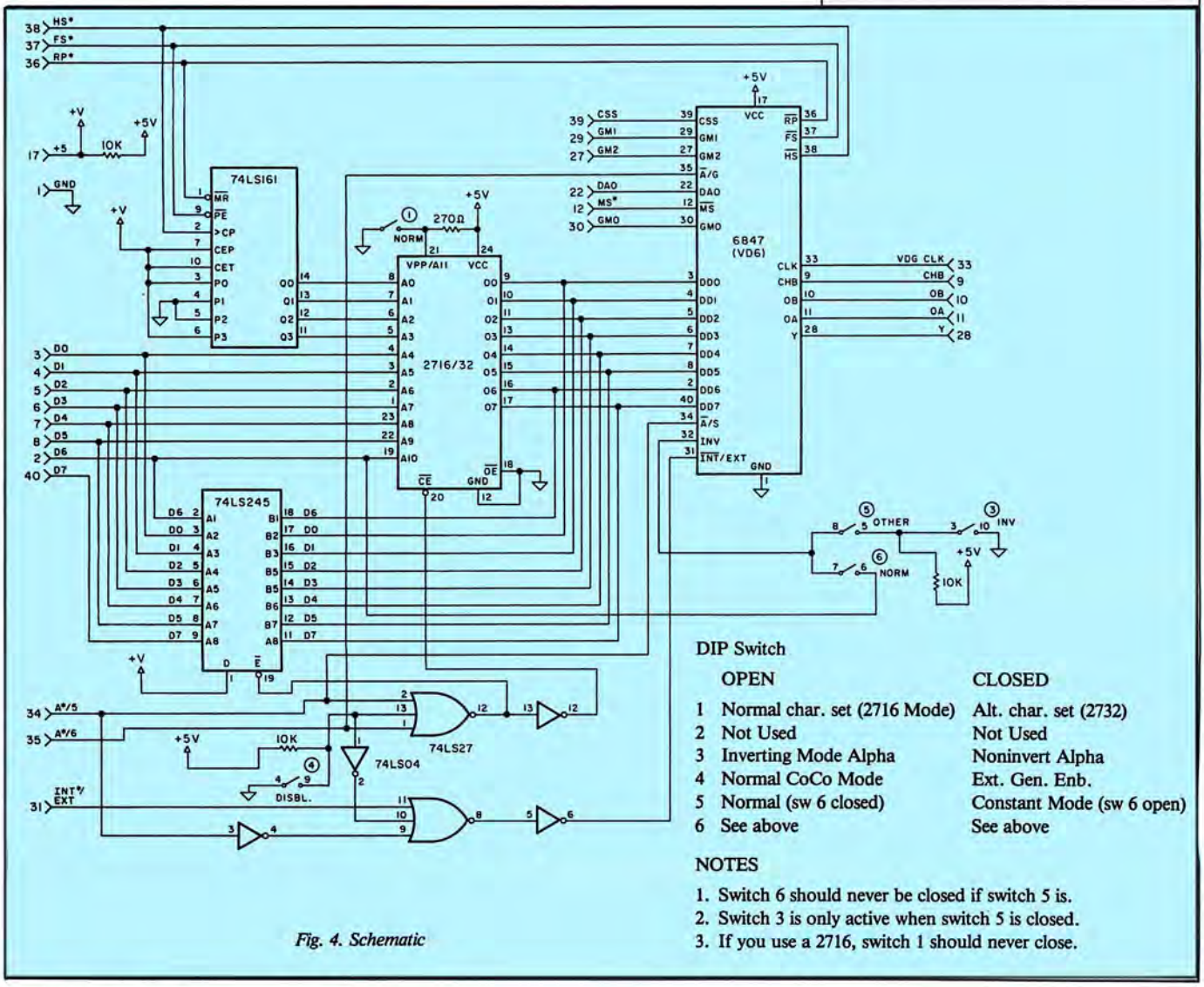


Fig. 4. Schematic

DIP Switch

	OPEN	CLOSED
1	Normal char. set (2716 Mode)	Alt. char. set (2732)
2	Not Used	Not Used
3	Inverting Mode Alpha	Noninvert Alpha
4	Normal CoCo Mode	Ext. Gen. Enb.
5	Normal (sw 6 closed)	Constant Mode (sw 6 open)
6	See above	See above

NOTES

1. Switch 6 should never be closed if switch 5 is.
2. Switch 3 is only active when switch 5 is closed.
3. If you use a 2716, switch 1 should never close.

Table 3. Values that Will Generate Lowercase Letters

```
1
2                                ;This is a table of the values that will generate lower case letters
3                                ;using an external character generator ROM for the 6847 VDG chip  .
4
5                                        section main,absolute
6
7                        0               ORG     0
8
9                                ;ch  @
10  00000000 00001C22            byte  0,0,1CH,22H,02H,1AH,2AH,2AH,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             021A2A2A
             1C000000
             FFFFFFFF
11                               ;ch a
12  00000010 00000000            byte  0,0,0,0,1CH,2,1EH,22H,1EH,0,0,0,0FFH,0FFH,0FFH,0FFH
             1C021E22
             1E000000
             FFFFFFFF
13                               ;ch b
14  00000020 00002020            byte  0,0,20H,20H,2CH,32H,22H,32H,2CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             2C322232
             2C000000
             FFFFFFFF
15                               ;ch c
16  00000030 00000000            byte  0,0,0,0,1CH,22H,20H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             1C222022
             1C000000
             FFFFFFFF
17                               ;ch d
18  00000040 00000202            byte  0,0,2,2,1AH,26H,22H,26H,1AH,0,0,0,0FFH,0FFH,0FFH,0FFH
             1A262226
             1A000000
             FFFFFFFF
19                               ;ch e
20  00000050 00000000            byte  0,0,0,0,1CH,22H,3EH,20H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             1C223E20
             1C000000
             FFFFFFFF
21                               ;ch f
22  00000060 0000040A            byte  0,0,4,0AH,8,1CH,8,8,8,0,0,0,0FFH,0FFH,0FFH,0FFH
             081C0808
             08000000
             FFFFFFFF
23                               ;ch g
24  00000070 00000000            byte  0,0,0,0,1AH,26H,26H,1AH,02H,22H,1CH,0,0FFH,0FFH,0FFH,0FFH
             1A26261A
             02221C00
             FFFFFFFF
25                               ;ch h
26  00000080 00002020            byte  0,0,20H,20H,2CH,32H,22H,22H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
             2C322222
             FFFFFFFF
27                               ;ch i
28  00000090 00000800            byte  0,0,8,0,18H,8,8,8,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             18080808
             1C000000
             FFFFFFFF
29                               ;ch j
30  000000A0 00000000            byte  0,0,0,0,2,0,2,2,2,22H,1CH,0,0FFH,0FFH,0FFH,0FFH
             02000202
             02221C00
             FFFFFFFF
31                               ;ch k
32  000000B0 00002020            byte  0,0,20H,20H,24H,28H,30H,28H,24H,0,0,0,0FFH,0FFH,0FFH,0FFH
             24283028
             24000000
             FFFFFFFF
33                               ;ch l
34  000000C0 00001808            byte  0,0,18H,8,8,8,8,8,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             08080808
             1C000000
             FFFFFFFF
35                               ;ch m
36  000000D0 00000020            byte  0,0,0,20H,36H,2AH,2AH,2AH,2AH,0,0,0,0FFH,0FFH,0FFH,0FFH
             362A2A2A
             2A000000
             FFFFFFFF
37                               ;ch n
38  000000E0 00000000            byte  0,0,0,0,1CH,32H,22H,22H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
             1C322222
             22000000
             FFFFFFFF
39                               ;ch o
40  000000F0 00000000            byte  0,0,0,0,1CH,22H,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             1C222222
             1C000000
             FFFFFFFF
41                               ;ch p
42  00000100 00000000            byte  0,0,0,0,2CH,32H,22H,32H,2CH,20H,20H,0,0FFH,0FFH,0FFH,0FFH
             2C322232
             2C202000
             FFFFFFFF
43                               ;ch q
44  00000110 00000000            byte  0,0,0,0,1AH,26H,22H,26H,1AH,2,2,0,0FFH,0FFH,0FFH,0FFH
             1A262226
             1A020200
             FFFFFFFF
45                               ;ch r
46  00000120 00000000            byte  0,0,0,0,2CH,32H,20H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
             2C322020
             20000000
             FFFFFFFF
47                               ;ch s
48  00000130 00000000            byte  0,0,0,0,1EH,20H,1CH,2,3CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             1E201C02
             3C000000
             FFFFFFFF
49                               ;ch t
50  00000140 00000808            byte  0,0,8,8,3EH,8,8,0AH,4,0,0,0,0FFH,0FFH,0FFH,0FFH
             3E08080A
             04000000
             FFFFFFFF
51                               ;ch u
52  00000150 00000000            byte  0,0,0,0,22H,22H,22H,26H,1AH,0,0,0,0FFH,0FFH,0FFH,0FFH
             22222226
             1A000000
             FFFFFFFF
53                               ;ch v
54  00000160 00000000            byte  0,0,0,0,22H,22H,22H,14H,8,0,0,0,0FFH,0FFH,0FFH,0FFH
             22222214
             08000000
             FFFFFFFF
55                               ;ch w
56  00000170 00000000            byte  0,0,0,0,22H,2AH,2AH,2AH,14H,0,0,0,0FFH,0FFH,0FFH,0FFH
             222A2A2A
             14000000
             FFFFFFFF
57                               ;ch x
58  00000180 00000000            byte  0,0,0,0,22H,14H,8,14H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
             22140814
             22000000
             FFFFFFFF
59                               ;ch y
60  00000190 00000000            byte  0,0,0,0,22H,22H,22H,1EH,2,22H,1CH,0,0FFH,0FFH,0FFH,0FFH
             2222221E
             02221C00
             FFFFFFFF
61                               ;ch z
62  000001A0 00000000            byte  0,0,0,0,3EH,4,8,10H,3EH,0,0,0,0FFH,0FFH,0FFH,0FFH
             3E040810
             3E000000
             FFFFFFFF
63                               ;ch [
64  000001B0 00001C10            byte  0,0,1CH,10H,10H,10H,10H,10H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             10101010
             1C000000
             FFFFFFFF
65                               ;ch \
66  000001C0 00002020            byte  0,0,20H,20H,10H,8,4,2,2,0,0,0,0FFH,0FFH,0FFH,0FFH
             10080402
             02000000
             FFFFFFFF
67                               ;ch ]
68  000001D0 00001C04            byte  0,0,1CH,4,4,4,4,4,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             04040404
             1C000000
             FFFFFFFF
69                                   ;ch  up arrow
70  000001E0 0000081C                 byte  0,0,8,1CH,2AH,8,8,8,8,0,0,0,0FFH,0FFH,0FFH,0FFH
             2A080808
             08000000
             FFFFFFFF
```

Table 3 continued

Table 3 continued

```
 71                             ;ch  left arrow
 72  000001F0 00000008          byte  0,0,0,8,10H,3EH,10H,8,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             103E1008
             00000000
             FFFFFFFF
 73                             ;ch space or dot for fill
 74  00000200 00000008          byte  0,0,0,8,0,0,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             00000000
             00000000
             FFFFFFFF
 75                             ;ch !
 76  00000210 00000808          byte  0,0,8,8,8,8,8,0,8,0,0,0,0FFH,0FFH,0FFH,0FFH
             08080800
             08000000
             FFFFFFFF
 77                             ;ch  "
 78  00000220 00001414          byte  0,0,14H,14H,14H,0,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             14000000
             00000000
             FFFFFFFF
 79                             ;ch #
 80  00000230 00001414          byte  0,0,14H,14H,3EH,14H,3EH,14H,14H,0,0,0,0FFH,0FFH,0FFH,0FFH
             3E143E14
             14000000
             FFFFFFFF
 81                             ;ch $
 82  00000240 0000081E          byte  0,0,8,1EH,28H,1CH,0AH,3CH,8,0,0,0,0FFH,0FFH,0FFH,0FFH
             281C0A3C
             08000000
             FFFFFFFF
 83                             ;ch %
 84  00000250 00003032          byte  0,0,30H,32H,4,8,10H,26H,6,0,0,0,0FFH,0FFH,0FFH,0FFH
             04081026
             06000000
             FFFFFFFF
 85                             ;ch &
 86  00000260 00001028          byte  0,0,10H,28H,28H,10H,2AH,24H,1AH,0,0,0,0FFH,0FFH,0FFH,0FFH
             28102A24
             1A000000
             FFFFFFFF
 87                             ;ch '
 88  00000270 00001818          byte  0,0,18H,18H,10H,20H,0,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             10200000
             00000000
             FFFFFFFF
 89                             ;ch (
 90  00000280 00000408          byte  0,0,4,8,10H,10H,10H,8,4,0,0,0,0FFH,0FFH,0FFH,0FFH
             10101008
             04000000
             FFFFFFFF
 91                             ;ch )
 92  00000290 00001008          byte  0,0,10H,8,4,4,4,8,10H,0,0,0,0FFH,0FFH,0FFH,0FFH
             04040408
             10000000
             FFFFFFFF
 93                             ;ch *
 94  000002A0 0000082A          byte  0,0,8,2AH,1CH,3EH,1CH,2AH,8,0,0,0,0FFH,0FFH,0FFH,0FFH
             1C3E1C2A
             08000000
             FFFFFFFF
 95                             ;ch +
 96  000002B0 00000008          byte  0,0,0,8,8,3EH,8,8,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             083E0808
             00000000
             FFFFFFFF
 97                             ;ch ,
 98  000002C0 00000000          byte  0,0,0,0,0,0,18H,18H,10H,20H,0,0,0FFH,0FFH,0FFH,0FFH
             00001818
             10200000
             FFFFFFFF
 99                             ;ch -
100  000002D0 00000000          byte  0,0,0,0,0,3EH,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             003E0000
             00000000
             FFFFFFFF
101                             ;ch .
102  000002E0 00000000          byte  0,0,0,0,0,0,0,18H,18H,0,0,0,0FFH,0FFH,0FFH,0FFH
             00000018
             18000000
             FFFFFFFF
103                             ;ch /
```

```
104  000002F0 00000202          byte  0,0,2,2,4,8,10H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
             04081020
             20000000
             FFFFFFFF
105                             ;ch 0
106  00000300 00001C20          byte  0,0,1CH,20H,26H,2AH,32H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             262A3222
             1C000000
             FFFFFFFF
107                             ;ch 1
108  00000310 00000818          byte  0,0,8,18H,8,8,8,8,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             08080808
             1C000000
             FFFFFFFF
109                             ;ch 2
110  00000320 00001C22          byte  0,0,1CH,22H,2,1CH,20H,20H,3EH,0,0,0,0FFH,0FFH,0FFH,0FFH
             021C2020
             3E000000
             FFFFFFFF
111                             ;ch 3
112  00000330 00001C22          byte  0,0,1CH,22H,2,0CH,2,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             020C0222
             1C000000
             FFFFFFFF
113                             ;ch 4
114  00000340 0000040C          byte    0,0,4,0CH,14H,24H,3EH,4,4,0,0,0,0FFH,0FFH,0FFH,0FFH
             14243E04
             04000000
             FFFFFFFF
115                             ;ch 5
116  00000350 00003E20          byte  0,0,3EH,20H,3CH,2,2,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             3C020222
             1C000000
             FFFFFFFF
117                             ;ch 6
118  00000360 00000C10          byte  0,0,0CH,10H,20H,3CH,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             203C2222
             1C000000
             FFFFFFFF
119                             ;ch 7
120  00000370 00003E02          byte  0,0,3EH,2,4,8,10H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
             04081020
             20000000
             FFFFFFFF
121                             ;ch 8
122  00000380 00001C22          byte  0,0,1CH,22H,22H,1CH,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
             221C2222
             1C000000
             FFFFFFFF
123                             ;ch 9
124  00000390 00001C22          byte  0,0,1CH,22H,22H,1EH,2,4,18H,0,0,0,0FFH,0FFH,0FFH,0FFH
             221E0204
             18000000
             FFFFFFFF
125                             ;ch :
126  000003A0 00000018          byte  0,0,0,18H,18H,0,18H,18H,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             18001818
             00000000
             FFFFFFFF
127                             ;ch ;
128  000003B0 00000018          byte  0,0,0,18H,18H,0,18H,18H,10H,20H,0,0,0FFH,0FFH,0FFH,0FFH
             18001818
             10200000
             FFFFFFFF
129                             ;ch <
130  000003C0 00000408          byte  0,0,4,8,10H,20H,10H,8,4,0,0,0,0FFH,0FFH,0FFH,0FFH
             10201008
             04000000
             FFFFFFFF
131                             ;ch >
132  000003D0 00000000          byte  0,0,0,0,3EH,0,3EH,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
             3E003E00
             00000000
             FFFFFFFF
133                             ;ch =
134  000003E0 00001008          byte  0,0,10H,8,4,2,4,8,10H,0,0,0,0FFH,0FFH,0FFH,0FFH
             04020408
             10000000
             FFFFFFFF
135                             ;ch ?
136  000003F0 00001C22          byte  0,0,1CH,22H,2,4,8,0,8,0,0,0,0FFH,0FFH,0FFH,0FFH
```

Table 3 continued

Table 3 continued

```
                02040800
                08000000
                FFFFFFFF
137                         ;ch  @
138 00000400    00001C22        byte  0,0,1CH,22H,2,1AH,2AH,2AH,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                021A2A2A
                1C000000
                FFFFFFFF
139                         ;ch A
140 00000410    00000814        byte  0,0,8,14H,22H,22H,3EH,22H,22H,0,0,0,0FFH,0FFH,0FQ ,0FFH
                22223E22
                22000000
                FFFFFFFF
141                         ;ch B
142 00000420    00003C12        byte  0,0,3CH,12H,12H,1CH,12H,12H,3CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                121C1212
                3C000000
                FFFFFFFF
143                         ;ch C
144 00000430    00001C22        byte  0,0,1CH,22H,20H,20H,20H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                20202022
                1C000000
                FFFFFFFF
145                         ;ch D
146 00000440    00003C12        byte  0,0,3CH,12H,12H,12H,12H,12H,3CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                12121212
                3C000000
                FFFFFFFF
147                         ;ch E
148 00000450    00003E20        byte  0,0,3EH,20H,20H,38H,20H,20H,3EH,0,0,0,0FFH,0FFH,0FFH,0FFH
                20382020
                3E000000
                FFFFFFFF
149                         ;ch F
150 00000460    00003E20        byte  0,0,3EH,20H,20H,38H,20H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
                20382020
                20000000
                FFFFFFFF
151                         ;ch G
152 00000470    00001E20        byte  0,0,1EH,20H,20H,26H,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                20262222
                1C000000
                FFFFFFFF
153                         ;ch H
154 00000480    00002222        byte  0,0,22H,22H,22H,3EH,22H,22H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                223E2222
                22000000
                FFFFFFFF
155                         ;ch I
156 00000490    00001C08        byte  0,0,1CH,8,8,8,8,8,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                08080808
                1C000000
                FFFFFFFF
157                         ;ch J
158 000004A0    00000202        byte  0,0,2,2,2,2,2,2,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                02020222
                1C000000
                FFFFFFFF
159                         ;ch K
160 000004B0    00002224        byte  0,0,22H,24H,28H,30H,28H,24H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                28302824
                22000000
                FFFFFFFF
161                         ;ch L
162 000004C0    00002020        byte  0,0,20H,20H,20H,20H,20H,20H,3EH,0,0,0,0FFH,0FFH,0FFH,0FFH
                20202020
                3E000000
                FFFFFFFF
163                         ;ch M
164 000004D0    00002236        byte  0,0,22H,36H,2AH,2AH,22H,22H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                2A2A2222
                22000000
                FFFFFFFF
165                         ;ch N
166 000004E0    00002232        byte  0,0,22H,32H,2AH,26H,22H,22H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                2A262222
                22000000
                FFFFFFFF
167                         ;ch O
168 000004F0    00001C22        byte  0,0,1CH,22H,22H,22H,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                22222222
                1C000000
                FFFFFFFF
169                         ;ch P
170 00000500    00003C22        byte  0,0,3CH,22H,22H,3CH,20H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
                223C2020
                20000000
                FFFFFFFF
171                         ;ch Q
172 00000510    00001C22        byte  0,0,1CH,22H,22H,22H,2AH,24H,1AH,0,0,0,0FFH,0FFH,0FFH,0FFH
                22222A24
                1A000000
                FFFFFFFF
173                         ;ch R
174 00000520    00003C22        byte  0,0,3CH,22H,22H,3CH,28H,24H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                223C2824
                22000000
                FFFFFFFF
175                         ;ch S
176 00000530    00001C22        byte  0,0,1CH,22H,20H,1CH,2,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                201C0222
                1C000000
                FFFFFFFF
177                         ;ch T
178 000005Q     00003E08        byte  0,0,3EH,8,8,8,8,8,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                08080808
                08000000
                FFFFFFFF
179                         ;ch U
180 00000550    00002222        byte  0,0,22H,22H,22H,22H,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                22222222
                1C000000
                FFFFFFFF
181                         ;ch V
182 00000560    00002222        byte  0,0,22H,22H,22H,14H,14H,8,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                22141408
                08000000
                FFFFFFFF
183                         ;ch W
184 00000570    00002222        byte  0,0,22H,22H,22H,22H,2AH,36H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                22222A36
                22000000
                FFFFFFFF
185                         ;ch X
186 00000580    00002222        byte  0,0,22H,22H,14H,8,14H,22H,22H,0,0,0,0FFH,0FFH,0FFH,0FFH
                14081422
                22000000
                FFFFFFFF
187                         ;ch Y
188 00000590    00002222        byte  0,0,22H,22H,14H,8,8,8,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                14080808
                08000000
                FFFFFFFF
189                         ;ch Z
190 000005A0    00003E02        byte  0,0,3EH,2,4,8,10H,20H,3EH,0,0,0,0FFH,0FFH,0FFH,0FFH
                04081020
                3E000000
                FFFFFFFF
191                         ;ch [
192 000005B0    00001C10        byte  0,0,1CH,10H,10H,10H,10H,10H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                10101010
                1C000000
                FFFFFFFF
193                         ;ch \
194 000005C0    00002020        byte  0,0,20H,20H,10H,8,4,2,2,0,0,0,0FFH,0FFH,0FFH,0FFH
                10080402
                02000000
                FFFFFFFF
195                         ;ch ]
196 000005D0    00001C04        byte  0,0,1CH,4,4,4,4,4,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                04040404
                1C000000
                FFFFFFFF
197                         ;ch  up arrow
198 000005E0    0000081C        byte  0,0,8,1CH,2AH,8,8,8,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                2A080808
                08000000
                FFFFFFFF
199                         ;ch  left arrow
200 000005F0    00000008        byte  0,0,0,8,10H,3EH,10H,8,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                103E1008
                00000000
```

Table 3 continued

Table 3 continued

```
                FFFFFFFF
201                         ;ch space
202  00000600 00000000             byte  0,0,0,0,0,0,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                00000000
                00000000
                FFFFFFFF
203                         ;ch !
204  00000610 00000808             byte  0,0,8,8,8,8,8,0,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                08080800
                08000000
                FFFFFFFF
205                         ;ch  "
206  00000620 00001414             byte  0,0,14H,14H,14H,0,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                14000000
                00000000
                FFFFFFFF
207                         ;ch #
208  00000630 00001414             byte  0,0,14H,14H,3EH,14H,3EH,14H,14H,0,0,0,0FFH,0FFH,0FFH,0FFH
                3E143E14
                14000000
                FFFFFFFF
209                         ;ch $
210  00000640 0000081E             byte  0,0,8,1EH,28H,1CH,0AH,3CH,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                281C0A3C
                08000000
                FFFFFFFF
211                         ;ch %
212  00000650 00003032             byte  0,0,30H,32H,4,8,10H,26H,6,0,0,0,0FFH,0FFH,0FFH,0FFH
                04081026
                06000000
                FFFFFFFF
213                         ;ch &
214  00000660 00001028             byte  0,0,10H,28H,28H,10H,2AH,24H,1AH,0,0,0,0FFH,0FFH,0FFH,0FFH
                28102A24
                1A000000
                FFFFFFFF
215                         ;ch '
Q16  00000670 00001818             byte  0,0,18H,18H,10H,20H,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                10200000
                00000000
                FFFFFFFF
217                         ;ch (
218  00000680 00000408             byte  0,0,4,8,10H,10H,10H,8,4,0,0,0,0FFH,0FFH,0FFH,0FFH
                10101008
                04000000
                FFFFFFFF
219                         ;ch )
220  00000690 00001008             byte  0,0,10H,8,4,4,4,8,10H,0,0,0,0FFH,0FFH,0FFH,0FFH
                04040408
                10000000
                FFFFFFFF
221                         ;ch *
222  000006A0 0000082A             byte  0,0,8,2AH,1CH,3EH,1CH,2AH,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                1C3E1C2A
                08000000
                FFFFFFFF
223                         ;ch +
224  000006B0 00000008             byte  0,0,0,8,8,3EH,8,8,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                083E0808
                00000000
                FFFFFFFF
225                         ;ch ,
226  000006C0 00000000             byte  0,0,0,0,0,0,18H,18H,10H,20H,0,0,0FFH,0FFH,0FFH,0FFH
                00001818
                10200000
                FFFFFFFF
227                         ;ch -
228  000006D0 00000000             byte  0,0,0,0,0,3EH,0,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                003E0000
                00000000
                FFFFFFFF
229                         ;ch .
230  000006E0 00000000             byte  0,0,0,0,0,0,0,18H,18H,0,0,0,0FFH,0FFH,0FFH,0FFH
                00000018
                18000000
                FFFFFFFF
231                         ;ch /
232  000006F0 00000202             byte  0,0,2,2,4,8,10H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
                04081020
                20000000
                FFFFFFFF
```

```
233                         ;ch 0
234  00000700 00001C20             byte  0,0,1CH,20H,26H,2AH,32H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                262A3222
                1C000000
                FFFFFFFF
235                         ;ch 1
236  00000710 00000818             byte  0,0,8,18H,8,8,8,8,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                08080808
                1C000000
                FFFFFFFF
237                         ;ch 2
238  00000720 00001C22             byte  0,0,1CH,22H,2,1CH,20H,20H,3EH,0,0,0,0FFH,0FFH,0FFH,0FFH
                021C2020
                3E000000
                FFFFFFFF
239                         ;ch 3
240  00000730 00001C22             byte  0,0,1CH,22H,2,0CH,2,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                020C0222
                1C000000
                FFFFFFFF
241                         ;ch 4
242  00000740 0000040C             byte  0,0,4,0CH,14H,24H,3EH,4,4,0,0,0,0FFH,0FFH,0FFH,0FFH
                14243E04
                04000000
                FFFFFFFF
243                         ;ch 5
244  00000750 00003E20             byte  0,0,3EH,20H,3CH,2,2,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                3C020222
                1C000000
                FFFFFFFF
245                         ;ch 6
246  00000760 00000C10             byte  0,0,0CH,10H,20H,3CH,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                203C2222
                1C000000
                FFFFFFFF
247                         ;ch 7
248  00000770 00003E02             byte  0,0,3EH,2,4,8,10H,20H,20H,0,0,0,0FFH,0FFH,0FFH,0FFH
                04081020
                20000000
                FFFFFFFF
249                         ;ch 8
250  00000780 00001C22             byte  0,0,1CH,22H,22H,1CH,22H,22H,1CH,0,0,0,0FFH,0FFH,0FFH,0FFH
                221C2222
                1C000000
                FFFFFFFF
251                         ;ch 9
252  00000790 00001C22             byte  0,0,1CH,22H,22H,1EH,2,4,18H,0,0,0,0FFH,0FFH,0FFH,0FFH
                221E0204
                18000000
                FFFFFFFF
253                         ;ch :
254  000007A0 00000018             byte  0,0,0,18H,18H,0,18H,18H,0,0,0,0,0FFH,0FFH,0FQ!,0FFH
                18001818
                00000000
                FFFFFFFF
255                         ;ch ;
256  000007B0 00000018             byte  0,0,0,18H,18H,0,18H,18H,10H,20H,0,0,0FFH,0FFH,0FFH,0FFH
                18001818
                10200000
                FFFFFFFF
257                         ;ch <
258  000007C0 00000408             byte  0,0,4,8,10H,20H,10H,8,4,0,0,0,0FFH,0FFH,0FFH,0FFH
                10201008
                04000000
                FFFFFFFF
259                         ;ch >
260  000007D0 00000000             byte  0,0,0,0,3EH,0,3EH,0,0,0,0,0,0FFH,0FFH,0FFH,0FFH
                3E003E00
                00000000
                FFFFFFFF
261                         ;ch =
262  000007E0 00001008             byte  0,0,10H,8,4,2,4,8,10H,0,0,0,0FFH,0FFH,0FFH,0FFH
                04020408
                10000000
                FFFFFFFF
263                         ;ch ?
264  000007F0 00001C22             byte  0,0,1CH,22H,2,4,8,0,8,0,0,0,0FFH,0FFH,0FFH,0FFH
                02040800
                08000000
                FFFFFFFF
265
266                               END
```

END

SUPER PRO KEYBOARD*

Great New Price!
Just $64.95

- Original layout—no unsupported keys.
- No special software required.
- Fast, simple installation—no soldering.
- Individually boxed with full instructions.
- Professional, low profile, finished appearance.
- U.S. made—highest quality, gold contacts.
- Smooth, responsive "Touch Typist" feel.
- Fits all 'D', 'E', and 'F' board models.

*Computers produced after approximately October 1982 require an additional plug adapter. Please add $4.95.

More Super Pro keyboards have been sold than any other brand for good reason. . .It is the best looking, best feeling keyboard available anywhere! The best buy for your money. Read what the reviewers have said:

Color Computer News, June '83

Mark Data Products is well known to us "longtimers"... Every bit as finished as if Tandy had done it...The Mark Data Super-Pro is your best buy...The one that is in my CoCo to stay...

Color Computer Magazine, June '83

The installation procedure is well detailed and quite simple...Has a professional feel, reacts well to the touch... has held up to some purposeful pounding...

Hot CoCo, August '83

Like putting leather upholstery in your Volkswagen...Very impressed with the appearance and performance...Could easily pass as original equipment...Installation is very simple...

Rainbow, April '83

A fine piece of hardware from Mark Data Products...It is super and it is professional too...If you are searching for a replacement keyboard, it is an excellent buy...

MD

Mark Data Products

24001 ALICIA PKWY., NO. 207 • MISSION VIEJO, CA 92691 • (714) 768-1551

All Orders: Please add $2.00 shipping and handling in the continental U.S. All others, add air shipping and $3.00 handling. California residents add 6% sales tax. Foreign orders please remit U.S. funds. Software authors—contact us for exciting program marketing details.

BY MARK D. GOODWIN

JOURNEY TO THE CENTER OF THE ROM—PART IX

As this series nears its end, ROM explorer Mark Goodwin delves into more of the Color Basic ROM. This month, he reveals areas including the LIST, LLIST, PRINT, MID$, and VAL commands.—*eds.*

Address correspondence to Mark Goodwin, Star Route 79, Box 103, Orland, ME 04472.

B6CF–B6F4 Color Basic MID$ Command

B6CF–B6D0	B = Default new string length
B6D1–B6D2	Save it
B6D3–B6D4	Get the next character
B6D5–B6D6	Is it a right parenthesis?
B6D7–B6D8	Jump if it's a right parenthesis
B6D9–B6DB	Check the syntax
B6DC–B6DD	Get the new string length
B6DE–B6DF	X = string address, A = string offset, and B = string offset
B6E0–B6E1	Display FC error message if string offset = 0
B6E2	B = default string length
B6E3	Decrement the string offset
B6E4–B6E5	String length < string offset?
B6E6–B6E7	Jump if the string length < the string offset
B6E8–B6E9	B = string offset
B6EA–B6EB	Subtract the string length from the string offset
B6EC	Make the result positive
B6ED–B6EE	Enough characters for the new string?
B6EF–B6F0	Jump if there aren't enough characters for the new string
B6F1–B6F2	B = new string length
B6F3–B6F4	Use the LEFT$ code

B6F5–B705 Get-the-String-Values Routine

B6F5–B6F7	Check the syntax
B6F8–B6F9	U = return address
B6FA–B6FB	X = string VARPTR
B6FC–B6FD	Save it
B6FE–B6FF	A = numeric value
B700–B701	B = numeric value
B702–B703	Clean up the stack
B704–B705	Return

B706–B715 Evaluate-Expression Routine

B706–B708	Display FC error message
B709–B70A	Get the next character
B70B–B70D	Evaluate the expression
B70E–B710	D = integer result
B711	D > 255?
B712–B713	Jump if D > 255
B714–B715	Get the next character

B716–B733 Color Basic VAL Command

B716–B718	X = string address and B = string length
B719–B71C	Jump if it's a null string
B71D–B71E	U = current ESP
B71F–B720	Save the string address as the current ESP
B721	X = end of the string pointer
B722–B723	A = last string character
B724–B725	Save the last string character, the end-of-the-string pointer, and the ESP
B726–B727	Last string character = 0
B728–B729	A = first string character
B72A–B72C	Convert the ASCII string to binary
B72D–B72E	Get the last string character, the end-of-the-string pointer, and the ESP
B72F–B730	Save the last character
B731–B732	Save the ESP
B733	Return

B734–B73C Evaluate-Expression Routine

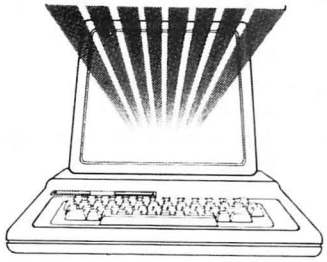
B734–B735	X = result of the expression
B736–B737	Save it
B738–B73A	Check the syntax
B73B–B73C	B = result of the expression

B73D–B74F Evaluate-Expression Routine

B73D–B73F	Evaluate the expression
B740–B741	A = (SF1)
B742–B743	Display FC error if it's negative
B744–B745	A = (EXP1)
B746–B747	Integer range?
B748–B749	Display FC error if it's not in the integer range

Listing continued

SUPER SCREEN



- **A big 51 character by 24 line screen.**
- **Full upper and lower case characters.**
- **Easily combine text with hi-res graphics.**
- **PRINT @ is completely functional on the big screen.**
- **The powerful ON ERROR GOTO is fully implemented.**
- **Auto-key repeat for greater keyboard convenience.**
- **Control codes for additional functions.**
- **Works with 16K, 32K or 64K computers.**
- **Available on disc or cassette.**
- **Works with extended and/or disc BASIC.**

51 CHARACTERS BY 24 LINE DISPLAY

Super Screen is a powerful, machine language program that significantly upgrades the performance and usefulness of 16K or greater, Extended and Disc Basic Color Computers. The standard Color Computer display screen is totally inadequate for serious, personal or business applications so Super Screen replaces it with a brand new, 51 character wide by 24 line screen including full upper and lower case characters. Instead of a confusing checkerboard appearance, you now have true lower case letters along with a screen that is capable of displaying 1224 characters. The difference is startling! Your computer takes on new dimensions and can easily handle lines of text that were simply too long and complex to display on the old screen.

COMBINE TEXT WITH HI-RES GRAPHICS

You can now write truly professional looking programs that combine text with hi-res graphics. Super Screen allows you to create graphics displays with the Basic LINE, DRAW and CIRCLE statements and then notate the graphics with descriptive text. You can even use PRINT @ if you wish for greater programming convenience. Super Screen's versatility will amaze you.

PRINT @ IS FULLY IMPLEMENTED

The PRINT @ statement is a valuable asset to the programmer when formatting text on the screen. The standard Color Computer will report an error if you specify a location higher than 511 but Super Screen allows locations all the way to 1223! You get a big screen and a powerful formatting tool as well. Of course, Super Screen also supports the CLS command allowing you to clear the big screen using standard Basic syntax.

ON ERROR GOTO

That's right! Super Screen gives you a full implementation of ON ERROR GOTO including the ERR and ERL functions. Now you can trap errors and take corrective action to prevent crashed programs and lost data using the same standard syntax as other computers. The ON ERROR GOTO capability overcomes a serious deficiency of Color Computer Basic and greatly improves your capability to handle sophisticated tasks. All well written, 'user friendly' programs use error trapping techniques and yours can too! Now that's power!

AUTO KEY REPEAT

No more frustration as you edit a long line in your Basic program; just hold the space bar down and automatically step to the desired position in the line. Need a line of asterisks? Hold the key down and auto repeat will give them to you. Those of you who spend many hours at your keyboard will appreciate this outstanding addition to Super Screen's long list of impressive capabilities.

CONTROL CODES FOR ADDITIONAL FUNCTIONS

Super Screen recognizes several special control code characters that allow selection of block or underline, solid or blinking cursor and other functions. You can 'Home Up' the cursor or you may erase from the cursor to the end of a line or to the end of the screen just like many other computers. These special codes give you an extra dimension of versatility and convenience that put Super Screen in a class by itself.

AND MORE GOOD NEWS...

Super Screen comes with complete, well detailed instructions and is available on cassette or disc. It adjusts automatically to any 16K or greater, Extended or Disc Basic Color Computer or TDP-100 and uses only 2K of memory in addition to the screen memory reserved during power up. Guaranteed to be the most frequently used program in your software library...once you use it, you won't be without it! Super Screen's low price will really please you; only $29.95 on cassette or $32.95 on disc!

64K Memory Expansion Kit
All parts and complete instructions
$64.95

NEW! SUPER BUG

Mark Data Products SUPER BUG is a powerful, relocatable machine code monitor program for your Coco. If you are a beginner, the program and documentation are an indispensable training aid, helping you to gain a better understanding of your Color Computer and machine code programming. If you are an accomplished computerist, SUPER BUG's capabilities, versatility and convenience will prove invaluable during programming and debugging.

SUPER BUG offers so many outstanding features that we are unable to list them all in this limited space. hex and alpha numeric memory display, modify, search and test; full printer support with baud rate and line feed select; up to 220 breakpoints; mini object code disassembler; 64K mode setup; decimal, hex and ascii code conversion routines and extensive documentation. Only $29.95 on cassette or $32.95 on disc.

ORDER ENTRY SYSTEM

The Mark Data Products sales order processing system will give a fast, efficient means to enter orders, print shipping papers and invoices, prepare sales reprots, and monitor receivables. The system automatically enhances the monitor screen to a 51 character by 24 line display. 32K of memory is required along with an 80-column printer, and one or more disc drives.

The MDP order entry system is a family of programs which operate interactively by means of a "menu" selection scheme. Up to 900 products may be defined and a single disc system can hold over 600 transactions. When the operator selects a task to be performed, the computer loads a program designed to handle that task from the system disc. The system disc contains all of the programs required to create, update and maintain data files and prepare the necessary paperwork including shipping and invoice forms, daily sales reports, a monthly (or other period) sales report and a receivables report.

The MDP system:

- Is accurate, user friendly and simple to use.
- Is easy to customize for specific user requirements.
- Produces a traceable invoice.
- Handles receivables as well as closed orders.
- In capable of future expandability.

This accounting software equals or exceeds higher priced packages for other computers and includes a detailed operating manual. For just $99.95.

ACCOUNTING SYSTEM

The Mark Data Products accounting system is ideal for the small businessman needing a fast, efficient means to process income and expenses, prepare detailed reports and maintain most of the information required at tax time. The system is a family of programs which operate by means of a "menu" selection scheme. When the operator selects a task to perform, the computer loads a program designed to handle that task from the system disc. The system disc contains all of the programs required to create, update and maintain data files and prepare the necessary accounting reports including a transaction journal, a P&L or income report, an interim or trial balance and a balance sheet.

Up to 255 separate accounts may be defined and a single disc system can hold over 1,400 transactions. This system automatically enhances the monitor screen to a 51 character by 24 line display. 32K of memory is required along with an 80-column printer and one or more disc drives.

The MDP system:

- Is accurate, user friendly and simple to use.
- Is easy to customize for specific user requirements.
- Immediately updates the chart of accounts.
- Provides an audit trail.
- Includes end of period procedures.
- Is capable of future expandability.

This order entry software equals or exceeds higher priced packages for other computers and includes a detailed operating manual. For just $99.95.

IMPORTANT NEW BOOKS

"Your Color Computer" by Doug Mosher. Over 300 pages of detailed information—A CoCo encyclopedia. $12.95.

"Programming the 6809" by Rodnay Zaks and William Labiak. One of the best 6809 machine language texts available—required reference material. $15.95.

WE STOCK SOFTLAW PRODUCTS

The VIP WRITER Text Processor is rated tops by Rainbow, Hot CoCo and Color Computer Magazine. After evaluation we rate it tops too. **Disc $59.95.**

MD

Mark Data Products

24001 ALICIA PKWY., NO. 207 • MISSION VIEJO, CA 92691 • (714) 768-1551

All Orders: Please add $2.00 shipping and handling in the continental U.S. All others, add air shipping and $3.00 handling. California residents add 6% sales tax. Foreign orders please remit U.S. funds. Software authors—Contact us for exciting program marketing details. We accept MasterCard and VISA. Distributed in Canada by Kelly Software

Listing continued

B74A–B74C	Convert FPAC1 to an integer
B74D–B74E	X = result
B74F	Return

B750–B756 Color Basic PEEK Command

B750–B751	X = address
B752–B753	B = byte value
B754–B756	Save B as the current result

B757–B75D Color Basic POKE Command

B757–B758	Evaluate the expression
B759–B75A	X = address
B75B–B75C	Save the byte
B75D	Return

B75E–B763 Color Basic LLIST Command

B75E–B75F	B = printer device number
B760–B761	Current device = printer
B762–B763	Get the next character

B764–B7C1 Color Basic LIST Command

B764–B765	Save the flags
B766–B768	Get the first line number
B769–B76B	X = address of the first line
B76C–B76D	Save it
B76E–B76F	Get the flags
B770–B771	Jump if it's the end of the Basic statement
B772–B773	Get the next character
B774–B775	Jump if it's the end of the Basic statement
B776–B777	Minus-sign token?
B778–B779	Jump if it's not a minus-sign token
B77A–B77B	Get the next character
B77C–B77D	Jump if it's the end of the Basic statement
B77E–B780	Get the second line number
B781–B782	Jump if it's the end of the Basic statement
B783	Return
B784–B786	U = default second line number
B787–B788	Save it
B789–B78A	Clean up the stack
B78B–B78C	X = address of the first line
B78D–B78F	Print a carriage return if necessary
B790–B792	Scan the keyboard
B793–B794	D = address of the next line
B795–B796	Jump if it's not the end of the program
B797–B799	Do CLOSE
B79A–B79B	Current device = video display
B79C–B79E	Jump to the command mode
B79F–B7A0	Save the address of the current line
B7A1–B7A2	D = current line number
B7A3–B7A5	Current line number > second line number?
B7A6–B7A7	Jump if the current line number > the second line number
B7A8–B7AA	Print the line number
B7AB–B7AD	Print a space
B7AE–B7AF	X = address of the current line
B7B0–B7B1	Untokenize the line
B7B2–B7B5	X = address of the next line
B7B6–B7B8	U = start of the buffer
B7B9–B7BA	A = next character
B7BB–B7BC	Jump if it's a null
B7BD–B7BF	Print it
B7C0–B7C1	Loop until the line has been printed

B7C2–B820 Untokenize Routine

B7C2–B7C4	Call the Extended Color Basic link
B7C5–B7C6	Bump to the start of the line
B7C7–B7CA	Y = start of the buffer
B7CB–B7CC	A = next character
B7CD–B7CE	Jump if it's the end of the line
B7CF–B7D0	Jump if it's a token
B7D1–B7D2	Colon?
B7D3–B7D4	Jump if it's not a colon
B7D5–B7D6	B = next character
B7D7–B7D8	ELSE token?
B7D9–B7DA	Jump if it's an ELSE token
B7DB–B7DC	Apostrophe token?
B7DD–B7DE	Jump if it's an apostrophe token
B7DF–B7E1	Ignore
B7E0–B7E1	A = exclamation mark
B7E2–B7E3	Save the character in the buffer
B7E4–B7E5	Loop until done
B7E6–B7E8	U = reserved-words-table pointer
B7E9–B7EA	Function prebyte?
B7EB–B7EC	Jump if it isn't a function prebyte
B7ED–B7EE	A = token
B7EF–B7F0	Point to the function table
B7F1–B7F2	Clear bit 7 of the token
B7F3–B7F4	Bump the reserved-words-table pointer
B7F5–B7F6	Reserved words left?
B7F7–B7F8	Jump if there aren't any more reserved words
B7F9–B7FA	Token in this list?
B7FB–B7FC	Jump if the token isn't in this list
B7FD–B7FE	Adjust the token
B7FF–B800	U = start of the reserved-words list
B801	Reserved word found?
B802–B803	Jump if the reserved word has been found
B804–B805	End of the reserved word?
B806–B807	Loop until it's the end of the reserved word
B808–B809	Loop
B80A–B80B	A = next character
B80C–B80D	Save it in the buffer
B80E–B80F	End of the reserved word?
B810–B811	Loop until it's the end of the reserved word
B812–B813	Loop
B814–B817	Buffer full?
B818–B819	Jump if the buffer is full
B81A–B81B	Mask the character
B81C–B81D	Save it in the buffer
B81E–B81F	Flag the end of the line
B820	Return

B821–B8F6 Tokenize Routine

B821–B823	Call the Extended Color Basic link
B824–B825	X = current ESP
B826–B828	U = start of the buffer
B829–B82A	Clear the tokenization flag
B82B–B82C	Clear the DATA flag
B82D–B82E	A = next character
B82F–B830	Jump if it's the end of the line
B831–B832	Tokenize?
B833–B834	Jump if tokenize
B835–B837	Character alphabetic?
B838–B839	Jump if it isn't alphabetic
B83A–B83B	Character < 0?
B83C–B83D	Jump if the character < 0
B83E–B83F	Numeric?
B840–B841	Jump if it's numeric
B842–B843	Clear the tokenization flag
B844–B845	Space?
B846–B847	Jump if it's a space
B848–B849	Save the character
B84A–B84B	Quote?
B84C–B84D	Jump if it's a quote
B84E–B84F	DATA?
B850–B851	Jump if not DATA
B852–B853	Save the character in the buffer
B854–B855	Jump if it's the end of the line
B856–B857	Colon?
B858–B859	Jump if it's a colon

Listing continued

Listing continued

B85A–B85B	Jump
B85C–B85D	Clear the next buffer location
B85E–B85F	Clear the next buffer location
B860–B861	D = end-of-the-buffer pointer
B862–B864	B = length of the tokenized line
B865–B867	X = start of the tokenized line – 1
B868–B869	Save it as the ESP
B86A	Return
B86B–B86C	Question mark?
B86D–B86E	Jump if it's not a question mark
B86F–B870	A = PRINT token
B871–B872	Jump
B873–B874	Apostrophe?
B875–B876	Jump if it's not an apostrophe
B877–B879	D = colon and apostrophe token
B87A–B87B	Save them in the buffer
B87C–B87D	Zero the stop value
B87E–B87F	A = next character
B880–B881	Jump if it's the end of the line
B882–B883	Character = stop value?
B884–B885	Jump if the character = stop value
B886–B887	Save the character in the buffer
B888–B889	Loop
B88A–B88B	Character < 0?
B88C–B88D	Jump if the character < 0
B88E–B88F	Less than a <?
B890–B891	Jump if it's less than a <
B892–B893	Decrement the buffer pointer
B894–B895	Save the buffer pointers
B896–B897	Clear the prebyte value
B898–B89A	U = start of the reserved-words-list table
B89B–B89C	Clear the token counter
B89D–B89E	Point to the next reserved-words block
B89F–B8A0	A = number of words in the list
B8A1–B8A2	Jump if there aren't any more reserved words
B8A3–B8A5	Y = start of the reserved-words list
B8A6–B8A7	X = buffer pointer
B8A8–B8A9	B = reserved-words character
B8AA–B8AB	Characters match?
B8AC–B8AD	Loop if they match
B8AE–B8AF	Bit 7 set?
B8B0–B8B1	Jump if bit 7 isn't set
B8B2–B8B3	Clean up the stack
B8B4–B8B5	U = tokenized buffer pointer
B8B6–B8B7	B = token value
B8B8–B8B9	A = prebyte value
B8BA–B8BB	Jump if there is a prebyte
B8BC–B8BD	ELSE token?
B8BE–B8BF	Jump if it isn't an ELSE token
B8C0–B8C1	A = colon
B8C2–B8C3	Save the prebyte and the token
B8C4–B8C5	Jump
B8C6–B8C7	Save the token
B8C8–B8C9	DATA token?
B8CA–B8CB	Jump if it isn't a DATA token
B8CC–B8CD	Set the DATA flag
B8CE–B8CF	REM token?
B8D0–B8D1	Jump if it's a REM token
B8D2–B8D3	Jump
B8D4–B8D6	U = start of the reserved-words-list table
B8D7–B8D8	Set the prebyte to FF
B8D9–B8DA	Jump if it's a function prebyte
B8DB–B8DC	Get the buffer pointers
B8DD–B8DE	A = next character

Listing continued

See List of Advertisers on page 99

Listing continued

B8DF–B8E0	Save it in the buffer
B8E1–B8E3	Alphabetic?
B8E4–B8E5	Jump if it's alphabetic
B8E6–B8E7	Set the tokenization flag
B8E8–B8E9	Jump
B8EA–B8EB	Bump the token counter
B8EC	End of the list?
B8ED–B8EE	Jump if it's the end of the list
B8EF–B8F0	Decrement the reserved-words pointer
B8F1–B8F2	B = next reserved-words character
B8F3–B8F4	Loop if bit 7 isn't set
B8F5–B8F6	Jump

B8F7–B99B Color Basic PRINT Command

B8F7–B8F8	Jump if it's the end of the Basic statement
B8F9–B8FA	Display the message
B8FB–B8FC	Current device = video display
B8FD	Return
B8FE–B8FF	Character = @?
B900–B901	Jump if it isn't an @
B902–B904	Update the cursor position
B905–B906	Jump
B907–B908	#?
B909–B90A	Jump if it isn't a #
B90B–B90D	Set the current device
B90E–B910	File OPEN?
B911–B912	Get the next character
B913–B914	Jump if it's the end of the Basic statement
B915–B917	Check the syntax
B918–B91A	Call the Extended Color Basic link
B91B–B91C	Jump if it's the end of the Basic statement
B91D–B91E	TAB(token?
B91F–B920	Jump if it's a TAB(token
B921–B922	Comma?
B923–B924	Jump if it's a comma
B925–B926	Semicolon?
B927–B928	Jump if it's a semicolon
B929–B92B	Evaluate the expression
B92C–B92D	A = NTF
B92E–B92F	Save it
B930–B931	Jump if it's a string
B932–B934	Convert FPAC1 to an ASCII string
B935–B937	Build a string entry
B938–B939	Display the string
B93A–B93B	Get the NTF
B93C–B93E	Figure the line position
B93F–B940	Cassette?
B941–B942	Jump if it isn't the cassette
B943–B944	Do a carriage return
B945–B946	Get the next character
B947–B948	Loop
B949	Done?
B94A–B94B	Jump if done
B94C–B94D	Get the next character
B94E–B94F	Comma?
B950–B951	Jump if it's a comma
B952–B953	Print a space
B954–B955	Get the next character
B956–B957	Loop
B958–B959	A = carriage return
B95A–B95B	Do a carriage return
B95C–B95E	Figure the line position
B95F–B960	Jump if cassette
B961–B962	A = number of characters in the line

Listing continued

Listing continued

B963–B964 Jump if it's not the start of the line
B965 Return
B966–B968 Figure the line position
B969–B96A Jump if cassette
B96B–B96C B = current line position
B96D–B96E Current line position < last comma position?
B96F–B970 Jump if the current line position < the last comma position
B971–B972 Do a carriage return
B973–B974 Loop
B975–B976 B = current line position
B977–B978 Subtract the comma field width
B979–B97A Loop until the number of spaces is figured
B97B Make it positive
B97C–B97D Use the TAB(code
B97E–B980 Evaluate the TAB(number
B981–B982 Next character = right parenthesis
B983–B986 Display SN error message if the next character < > right parenthesis
B987–B989 Figure the line position
B98A–B98B Figure the number of spaces
B98C–B98D Jump if the line position > the tab
B98E–B98F Cassette?
B990–B991 Jump if it's the cassette
B992–B993 Print a space
B994 All spaces printed?
B995–B996 Loop until all the spaces have been printed
B997–B998 Get the next character
B999–B99B Loop

B99C–B9AB Display-Message Routine

B99C–B99E Build a string entry
B99F–B9A1 X = string address and B = string length
B9A2 Bump the string length
B9A3 All characters printed?
B9A4–B9A5 Jump if all the characters have been printed
B9A6–B9A7 A = next character
B9A8–B9A9 Print it
B9AA–B9AB Loop

B9AC–B9B3 Output Routine

B9AC–B9AD A = space
B9AE–B9B0 Ignore
B9AF–B9B0 A = question mark
B9B1–B9B3 Print the character

B9B4–B9B8 FPAC1 = FPAC1 + .5

B9B4–B9B6 X = floating-point constant .5 pointer
B9B7–B9B8 Jump

B9B9–B9BB FPAC1 = (X) – FPAC1

B9B9–B9BB Move the floating-point value (X) into FPAC2

B9BC–B9C1 FPAC1 = FPAC2 – FPAC1

B9BC–B9BD Invert (SF1)
B9BE–B9BF Invert (SF)
B9C0–B9C1 Jump

B9C2–B9C4 FPAC1 = (X) + FPAC1

B9C2–B9C4 Move the floating-point value (X) into FPAC2

B9C5–BA17 FPAC1 = FPAC2 + FPAC1

B9C5 FPAC1 = 0?
B9C6–B9C9 Jump if FPAC1 = 0
B9CA–B9CC X = start of FPAC2
B9CD–B9CE B = (EXP2)
B9CF FPAC2 = 0?
B9D0–B9D1 Jump if FPAC2 = 0
B9D2–B9D3 Compare the exponents
B9D4–B9D5 Jump if (EXP1) = (EXP2)
B9D6–B9D7 Jump if (EXP1) > (EXP2)
B9D8–B9D9 Save (EXP2) as (EXP1)
B9DA–B9DB A = (SF2)
B9DC–B9DD Save it as (SF1)
B9DE–B9E0 X = start of FPAC1
B9E1 Make the difference between the exponents negative
B9E2–B9E3 Difference < = eight places
B9E4–B9E5 Jump if the difference < = eight places
B9E6 Zero A
B9E7–B9E8 Shift (MSB1)
B9E9–B9EB Shift FPAC1 until it lines up with FPAC2
B9EC–B9ED B = (SF)
B9EE–B9EF Jump if the result will be positive
B9F0–B9F1 Invert (MSB1)
B9F2–B9F3 Invert (NMSB1)
B9F4–B9F5 Invert (NNMSB1)
B9F6–B9F7 Invert (LSB1)
B9F8 Invert (RB)
B9F9–B9FA Add carry to (RB)
B9FB–B9FC Save it
B9FD–B9FE A = (LSB1)
B9FF–BA00 A = (LSB1) + (LSB2)
BA01–BA02 Save it as (LSB1)
BA03–BA04 A = (NMSB1)
BA05–BA06 A = (NNMSB1) + (NNMSB2) + carry
BA07–BA08 Save it as (NNMSB1)
BA09–BA0A A = (NMSB1)
BA0B–BA0C A = (NMSB1) + (NMSB2) + carry
BA0D–BA0E Save it as (NMSB1)
BA0F–BA10 A = (MSB1)
BA11–BA12 A = (MSB1) + (MSB2) + carry
BA13–BA14 Save it as (MSB1)
BA15 Check the sign
BA16–BA17 Jump if the result is positive

BA18–BA3E Shift-FPAC1 Routine

BA18–BA19 Jump if FPAC1 overflowed
BA1A–BA1B Make the result negative
BA1C Bit counter = 0
BA1D–BA1E A = (MSB1)
BA1F–BA20 Jump if (MSB1) < > 0
BA21–BA22 A = (NMSB1)
BA23–BA24 Save it as (MSB1)
BA25–BA26 A = (NNMSB1)
BA27–BA28 Save it as (NMSB1)
BA29–BA2A A = (LSB1)
BA2B–BA2C Save it as (NNMSB1)
BA2D–BA2E A = (RB)
BA2F–BA30 Save it as (LSB1)
BA31–BA32 (RB) = 0
BA33–BA34 Bump the bit counter for the number of shifts
BA35–BA36 All bits shifted?
BA37–BA38 Loop if all the bits haven't been shifted
BA39 Zero A
BA3A–BA3B Save it as (EXP1)
BA3C–BA3D Save it as (SF1)
BA3E Return

BA3F–BA43 Part of Addition Routine

BA3F–BA40 Line up the values
BA41 Zero B
BA42–BA43 Jump

BA44–BA78 Shift FPAC1 Routine

BA44 Bump the bit counter
BA45–BA46 Shift (RB)
BA47–BA48 Shift (LSB1)

Listing continued

Listing continued

BA49-BA4A	Shift (NNMSB1)
BA4B-BA4C	Shift (NMSB1)
BA4D-BA4E	Shift (MSB1)
BA4F-BA50	Loop if bit 7 clear
BA51-BA52	A = (EXP1)
BA53-BA54	Save the bit counter
BA55-BA56	Subtract the number of shifts
BA57-BA58	Save it as (EXP1)
BA59-BA5A	Jump if the exponent is too small
BA5B-BA5D	Ignore
BA5C-BA5D	Jump if done
BA5E-BA5F	Shift (RB)
BA60-BA61	Zero A
BA62-BA63	Save it as (RB)
BA64-BA65	Jump
BA66-BA67	Bump (EXP1)
BA68-BA69	Jump if overflow
BA6A-BA6B	Shift (MSB1)
BA6C-BA6D	Shift (NMSB1)
BA6E-BA6F	Shift (NNMSB1)
BA70-BA71	Shift (LSB1)
BA72-BA73	Jump if done
BA74-BA75	Bump FPAC1
BA76-BA77	Jump if (MSB1) and (NMSB1) = 0
BA78	Return

BA79-BA91 Two's-Complement Routine

BA79-BA7A	Invert (SF1)
BA7B-BA7C	Invert (MSB1)
BA7D-BA7E	Invert (NMSB1)
BA7F-BA80	Invert (NNMSB1)
BA81-BA82	Invert (LSB1)
BA83-BA84	X = (NNMSB1) and (LSB1)
BA85-BA86	Bump them
BA87-BA88	Save them
BA89-BA8A	Jump if they aren't equal to zero
BA8B-BA8C	X = (MSB1) and (NMSB1)
BA8D-BA8E	Bump them
BA8F-BA90	Save them
BA91	Return

BA92-BA96 OV-Error Routine

BA92-BA93	B = OV error code
BA94-BA96	Display OV error message

BA97-BAC4 Shift Floating-Point Value at (X)

BA97-BA99	(X) = multiplication total
BA9A-BA9B	A = (LSB)
BA9C-BA9D	Save it as (RB)
BA9E-BA9F	A = (NNMSB)
BAA0-BAA1	Save it as (LSB)
BAA2-BAA3	A = (NMSB)
BAA4-BAA5	Save it as (NNMSB)
BAA6-BAA7	A = (MSB)
BAA8-BAA9	Save it as (NMSB)
BAAA-BAAB	A = shift value
BAAC-BAAD	Save it as (MSB)
BAAE-BAAF	Adjust the bit counter
BAB0-BAB1	Loop if the shifts left > eight places
BAB2-BAB3	A = (RB)
BAB4-BAB5	Adjust the bit counter
BAB6-BAB7	Jump if done
BAB8-BAB9	Shift (MSB)
BABA-BABB	Shift (NMSB)
BABC-BABD	Shift (NNMSB)
BABE-BABF	Shift (LSB)
BAC0	Shift (RB)
BAC1	Shifting done?
BAC2-BAC3	Loop until the shifting is done
BAC4	Return

BAC5-BAC9 Floating-Point Constant Equal to 1

BACA-BB2E FPAC1 = (X) * FPAC1

BACA-BACB	Move floating-point value (X) into FPAC2
BACC-BACD	Jump if FPAC1 = 0
BACE-BACF	Adjust (EXP1) and (SF1)
BAD0-BAD1	Zero A
BAD2-BAD3	MSB of total = 0
BAD4-BAD5	NMSB of total = 0
BAD6-BAD7	NNMSB of total = 0
BAD8-BAD9	LSB of total = 0
BADA-BADB	B = (LSB1)
BADC-BADD	Multiply FPAC2 by (LSB1)
BADE-BADF	B = (RB)
BAE0-BAE1	Save it for RND
BAE2-BAE3	B = (NNMSB1)
BAE4-BAE5	Multiply FPAC2 by (NNMSB1)
BAE6-BAE7	B = (RB)
BAE8-BAE9	Save it for RND
BAEA-BAEB	B = (NMSB1)
BAEC-BAED	Multiply FPAC2 by (NMSB1)
BAEE-BAEF	B = (RB)
BAF0-BAF1	Save it for RND
BAF2-BAF3	B = (MSB1)
BAF4-BAF5	Multiply FPAC2 by (MSB1)
BAF6-BAF7	B = (RB)
BAF8-BAF9	Save it for RND
BAFA-BAFC	Move the total into FPAC1
BAFD-BAFF	Normalize the result
BB00-BB01	Jump if it's equal to zero
BB02	Invert A
BB03-BB04	A = MSB of the total
BB05	Put the next bit into carry
BB06-BB07	Jump if done
BB08-BB09	Jump if it's not set
BB0A-BB0B	A = LSB of the total
BB0C-BB0D	A = LSB of the total + (LSB2)
BB0E-BB0F	Save it
BB10-BB11	A = NNMSB of the total
BB12-BB13	A = NNMSB of the total + (NNMSB2) + carry
BB14-BB15	Save it
BB16-BB17	A = NMSB of the total
BB18-BB19	A = NMSB of the total + (NMSB2) + carry
BB1A-BB1B	Save it
BB1C-BB1D	A = MSB of the total
BB1E-BB1F	A = MSB of the total + (MSB2) + carry
BB20	Shift it
BB21-BB22	Save it
BB23-BB24	Shift the NMSB of the total
BB25-BB26	Shift the NNMSB of the total
BB27-BB28	Shift the LSB of the total
BB29-BB2A	Shift it into (RB)
BB2B	Zero A
BB2C-BB2D	Loop until done
BB2E	Return

BB2F-BB47 Move (X) to FPAC2

BB2F-BB30	D = MSB and NMSB
BB31-BB32	Save the MSB as (SF2)
BB33-BB34	Set bit 7 of the MSB
BB35-BB36	Save (MSB2) and (NMSB2)
BB37-BB38	B = (SF2)
BB39-BB3A	Combine (SF1) and (SF2)
BB3B-BB3C	Save it
BB3D-BB3E	D = NNMSB and LSB
BB3F-BB40	Save (NNMSB2) and (LSB2)
BB41-BB42	A = EXP
BB43-BB44	Save (EXP2)
BB45-BB46	B = (EXP1)
BB47	Return ■

The Joystick that sets you free!

The one-hand operation of this fantastic new joystick will truly set you free and increase the pleasure of playing your favorite video games. The smoothness and responsiveness of this unique joystick that operates completely without a base is something to be experienced. Available direct from us or from your independent computer retail store. (See below)

$49.95 suggested retail

STOP changing Printer and Modem Cables! Our **Parallel Printer Interface** provides **Switch Selectable Printer** or **Modem** operations for both CoCo and MC10. It features switchable baud rates from 300 to 9600. It comes complete with power supply, modem cable and **"Centronics"** type printer cable. For Basic 1.1 and later revisions.

Available direct from us or from your independent computer retail store. (See below)

Only $89.95 suggested retail

pbh Computer Products, Inc.
P. O. Drawer 55868
Houston, Texas 77055
713/956-0207

↙20

When ordering direct from **PBH** please enclose $3.00 per item for shipping.

Stocking Distributors

Compukit
Houston, TX. 77059

Spectrum Projects
Woodhaven, N.Y. 11421

Authorized Dealers

Endicott Computer
Software & Accessories
Huntsville, AL. 35801

The Computer Store
Jasper, IN. 47456

The Software Connection
Ft. Lauderdale, FL. 33319

Colorware, Inc.
Woodhaven, N.Y. 11421

TRS TECH
Computer Services
Houston, TX. 77033

Computers & More
Huntsville, TX. 77340

The Photo Shop Radio Shack
Wilmar, MN. 56201

Patterson Electronics
Mountain View, AR. 72560

Computers, Etc.
Austin, TX. 78745

Cinsoft
Cincinnati, OH. 45237

EDC Industries
Los Angeles, CA. 90042

Sound Center Radio Shack
Whiterock, N.M. 87644
& Los Alamos, N.M. 87544

Chips, Incorporated
Atlanta, GA. 30340

Computer Associates
West Fargo, N.Dak. 58078

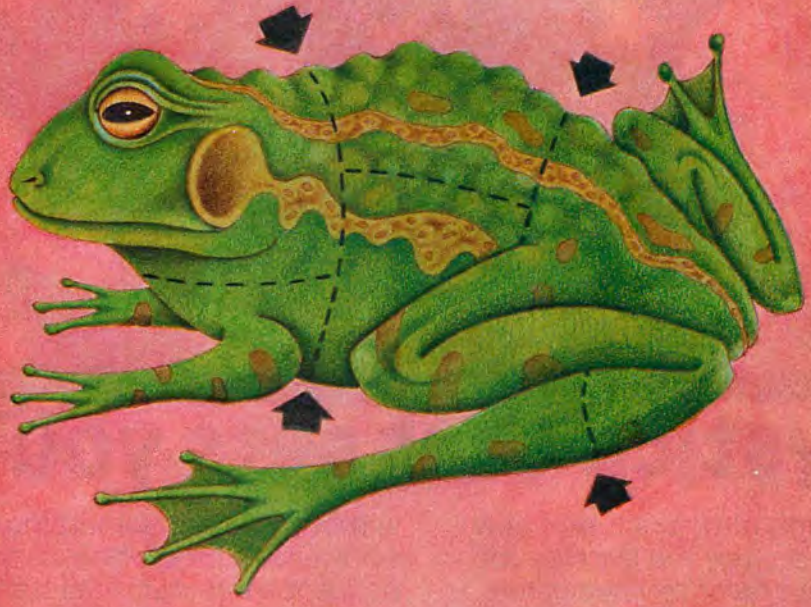
Computer Plus, Inc.
Littleton, MA. 01460

Turtle Micro Ware
East Lime, CT. 06333

↙ See List of Advertisers on page 99

TUTORIAL

BY MIKE MEEHAN



ANATOMY OF AN ASSEMBLY-LANGUAGE GAME—PART II

Continue the exploration of an Assembly-language game with a look at what makes this Croaker hop.

The June issue of *HOT CoCo* introduced you to techniques used in an Assembly-language game. This month you enter all the data for Croaker, and I talk about the Color Computer's memory map. If you followed last month's article, you should have a copy of the title page and an editor/assembler ready.

The Color Computer's memory is divided into several sections, a listing of which is called a memory map. (See Table 1.) With the exception of input/output, each of these areas is labeled either RAM or ROM.

RAM (random-access memory) can either be written to or read from. ROM (read-only memory), on the other hand, can only be read from. ROM stores all of Basic and any cartridges you own, while RAM stores your Basic or Assembly-language programs.

The Basic commands to write to and read from memory are POKE and

PEEK. POKE a number from 0–255 into any area of user RAM. For example, to POKE a value of 30 into 2000, you use the Basic command POKE 2000,30. You can then use the command PRINT PEEK(2000) to examine the contents of location 2000.

Try this on RAM, where the numbers you read back should be the numbers you POKEd in, and then on some of the ROM areas. Chances are the number you read back will be different than the one you POKEd in the ROM areas. If you try several other numbers, the value you read back should remain constant.

If you have 64K, all the programs that let you access the 64K turn ROM areas into RAM and store Basic in them. Therefore, if you are running 64K, the previous test for ROM fails and you read back the same number you POKEd in.

Think of direct page RAM as a scratch pad for Basic and the microprocessor, storing the values they need. If you need more specific information on the use of each memory location, refer to your Color Basic manual or buy one of the full memory maps available from software vendors.

The area from 1024–1535 stores the video screen. By changing the contents of this area, you can change the character screen. For example, POKE a value of 255 into 1535. A pink block should appear in the lower right corner of the screen. You can experiment with this to get an idea of how the screen works.

On a 16K system, your user RAM (solely for user's purposes) is from 1536–16383, and with 32K it is from 1536–32767. 64K is more complicated. The user RAM in Basic is from 1536–32767. Otherwise, in machine language your user RAM is from 1536–65279. Croaker is stored in user RAM, where the graphics are done.

Basic occupies the area from 32768–65279 most of the time unless you are running 64K and are not using Basic. In that case the area becomes user RAM.

The input/output area controls the I/O functions such as cassette, sound, and graphics mode. Since it is made up mostly of switches that are either on or off, it is marked neither RAM nor ROM. In some addresses, separate bits control separate functions, and again, a full memory map will give you more information on the use of each location.

Croaker uses a music routine that simulates organ music in four-part harmony. It first appeared in the March 1982 issue of *68 Micro Journal* (p. 35) and then in the July 1982 issue of *Color Computer News* (p. 16) and is based on an algorithm by Hal Chamberlin (*Byte*, September 1977).

The actual routine appears next month, but this month you enter the data for all the songs, and the wave and note tables for the routine. Data for the music is entered in the following format:

length of note
1st note 2nd note
3rd note 4th note

The only real song is the title music labeled SONG1. The rest are actually sound effects for the game. Croaker uses the music routine for all its sound.

Codes for all the prompts you receive in the game are also in the data. Because the game uses an inverse screen (dark red), you must enter the codes manually rather than putting the words between slashes. For example, the inverse code for A is 1. POKE a 1 into 1535 and an inverse A appears in the lower right corner of the screen.

The codes for a set of logs, turtles, and turtles disappearing underwater are also entered, divided into an 8-across, 10-down matrix. If the data looks like this:

```
1 FDB $AAAA
2 FDB $AAAA
3 FDB $AAAA
4 FDB $AAAA
5 FDB $A000
6 FDB $0000
7 FDB $0000
8 FDB $02AA
```

it will be POKEd onto the high-resolution screen as:

```
AA AA AA AA AA AA AA AA
A0 00 00 00 00 00 02 AA
```

The actual data continues until the 8-by-10 matrix is completed.

The codes for frogs, divided into a 2-by-10 matrix, are also entered. If the data looks like this:

```
1 FDB $FFFF
2 FDB $75DF
```

it is POKEd onto the hi-res screen as:

```
FF FF
75 DF
```

and continues until the 2-by-10 matrix is completed.

You must enter a total of 24 different frogs, divided into three major sections in the data: frogs on a red background, white background, and blue background. There are eight frogs in each color background. One faces up, one down, one left, and one right. There is also a set in the middle of a jump, facing up, down, left, and right.

These give more realistic movement during the game. The frogs appear on different backgrounds so when a frog jumps from one color background to another, there is a frog in memory with the corresponding background color that can be transposed onto the screen.

You also enter a skull on a white background, one on a blue background, and one last frog, all of which use the frogs' 2-by-10 matrix. The skulls are used when one of the frogs croaks (pun intended). The last frog is the one placed in the base after making it through the obstacles.

You finally enter six cars of different colors, all formed in a 4-by-10 matrix. If the data appears as:

```
1 FDB $0000
2 FDB $0000
3 FDB $0FC3
4 FDB $F0FC
```

it will be POKEd onto the hi-res screen as:

"Congratulations! You have passed the endurance test. This is by far the longest program of the series."

Addresses (decimal)	Contents
0–1023	Direct page RAM
1024–1535	Video memory RAM
1536–16383	User RAM (16K)
1536–32767	User RAM (32K)
32768–40959	Extended Basic ROM
40960–49151	Color Basic ROM
49152–65279	Disk Basic ROM or cartridge ROM
65280–65535	Input/Output

Table 1. Color Computer Memory Map

System Requirements

32K RAM
Editor/Assembler

```
00  00  00  00
0F  C3  F0  FC
```

and continues until the 4-by-10 matrix is completed.

Congratulations! You have passed the endurance test. This is by far the longest program of the series, and next month you will enter and read about some little subroutines. I can provide you with a cassette copy of the Croaker program for $10. Questions and comments are always welcome, just include a self-addressed, stamped envelope if you want a response. ■

Address correspondence to Michael Meehan, 1300 Fairfield Drive, Clearwater, FL 33546.

Program Listing. Croaker Data

```
00100 *************************
00110 *********CROAKER*********
00120 *****BY: MIKE MEEHAN*****
00130 *****COPYRIGHT 1983******
00140 *COLOR HORIZONS SOFTWARE*
00150 *************************
00160 *********PART TWO********
00170 *************************
00180          ORG     $1E00
00190 SONG1    FCB     $10     CODES FOR
00200          FDB     $4800   TITLE MUSIC
00210          FDB     $0000
00220          FCB     $09
00230          FDB     $4848
00240          FDB     $4040
00250          FCB     $09
00260          FDB     $4800
00270          FDB     $0000
00280          FCB     $09
00290          FDB     $4848
00300          FDB     $4040
00310          FCB     $09
00320          FDB     $4800
00330          FDB     $0000
00340          FCB     $09
00350          FDB     $4848
00360          FDB     $4040
00370          FCB     $09
00380          FDB     $0000
00390          FDB     $0000
00400          FCB     $10
00410          FDB     $4800
00420          FDB     $0000
00430          FCB     $09
00440          FDB     $4848
00450          FDB     $4040
00460          FCB     $09
00470          FDB     $4800
00480          FDB     $0000
00490          FCB     $09
00500          FDB     $4848
00510          FDB     $4040
00520          FCB     $09
00530          FDB     $4800
00540          FDB     $0000
00550          FCB     $09
00560          FDB     $4848
00570          FDB     $4040
00580          FCB     $07
00590          FDB     $0000
00600          FDB     $0000
00610          FCB     $0B
00620          FDB     $4C00
00630          FDB     $0000
00640          FCB     $07
00650          FDB     $0000
00660          FDB     $0000
00670          FCB     $0B
00680          FDB     $4C00
00690          FDB     $0000
00700          FCB     $07
00710          FDB     $0000
00720          FDB     $0000
00730          FCB     $0B
00740          FDB     $4A00
00750          FDB     $0000
00760          FCB     $07
00770          FDB     $0000
00780          FDB     $0000
00790          FCB     $0B
00800          FDB     $4A00
00810          FDB     $0000
00820          FCB     $07
00830          FDB     $0000
00840          FDB     $0000
00850          FCB     $29
00860          FDB     $4600
00870          FDB     $0000
00880          FCB     $06
00890          FDB     $0000
00900          FDB     $0000
00910          FCB     $0B
00920          FDB     $4C00
00930          FDB     $0000
00940          FCB     $07
00950          FDB     $0000
00960          FDB     $0000
00970          FCB     $0B
00980          FDB     $4C00
00990          FDB     $0000
01000          FCB     $07
01010          FDB     $0000
01020          FDB     $0000
01030          FCB     $0B
01040          FDB     $4A00
01050          FDB     $0000
01060          FCB     $07
01070          FDB     $0000
01080          FDB     $0000
01090          FCB     $0B
01100          FDB     $4A00
01110          FDB     $0000
01120          FCB     $07
01130          FDB     $0000
01140          FDB     $0000
01150          FCB     $0B
01160          FDB     $4600
01170          FDB     $0000
01180          FCB     $07
01190          FDB     $0000
01200          FDB     $0000
01210          FCB     $0B
01220          FDB     $4600
01230          FDB     $0000
01240          FCB     $07
01250          FDB     $0000
01260          FDB     $0000
01270          FCB     $0B
01280          FDB     $5400
01290          FDB     $0000
01300          FCB     $07
01310          FDB     $0000
01320          FDB     $0000
01330          FCB     $0B
01340          FDB     $5400
01350          FDB     $0000
01360          FCB     $07
01370          FDB     $0000
01380          FDB     $0000
01390          FCB     $0B
01400          FDB     $5000
01410          FDB     $0000
```

Listing continued

Listing continued

```
01420            FCB    $07
01430            FDB    $0000
01440            FDB    $0000
01450            FCB    $0B
01460            FDB    $4C00
01470            FDB    $0000
01480            FCB    $07
01490            FDB    $0000
01500            FDB    $0000
01510            FCB    $0B
01520            FDB    $4A00
01530            FDB    $0000
01540            FCB    $07
01550            FDB    $0000
01560            FDB    $0000
01570            FCB    $0B
01580            FDB    $4600
01590            FDB    $0000
01600            FCB    $07
01610            FDB    $0000
01620            FDB    $0000
01630            FCB    $30
01640            FDB    $4200
01650            FDB    $0000
01660            FCB    $00
01670 SONG2      FCB    $07     CODES FOR
01680            FDB    $1818   ALARM
01690            FDB    $1A1A
01700            FCB    $04
01710            FDB    $0000
01720            FDB    $0000
01730            FCB    $07
01740            FDB    $1818
01750            FDB    $1A1A
01760            FCB    $04
01770            FDB    $0000
01780            FDB    $0000
01790            FCB    $07
01800            FDB    $1818
01810            FDB    $1A1A
01820            FCB    $00
01830 SONG3      FCB    $02     CODES FOR
01840            FDB    $4040   FROG MOVEMEN
01850            FDB    $4040   MUSIC
01860            FCB    $02
01870            FDB    $4242
01880            FDB    $4242
01890            FCB    $00
01900 SONG4      FCB    $02
01910            FDB    $4444
01920            FDB    $4444
01930            FCB    $02
01940            FDB    $4646
01950            FDB    $4646
01960            FCB    $00
01970 SONG5      FCB    $01     BLANK CODES
01980            FDB    $0000   FOR LATER
01990            FDB    $0000   USE
02000            FCB    $00
02010 PLAYER     FDB    $100C   CODES FOR
02020            FDB    $0119   'PLAYER'
02030            FDB    $0512
02040 NPLAY      FDB    $0E15   CODES FOR
02050            FDB    $0D02   'NUMBER OF
02060            FDB    $0512   PLAYERS?'
02070            FDB    $200F
02080            FDB    $0620
02090            FDB    $100C
02100            FDB    $0119
02110            FDB    $0512
02120            FDB    $133F
02130 PRESS      FDB    $1012   CODES FOR
02140            FDB    $0513   'PRESS JOYSTICK
02150            FDB    $1320   BUTTON.'
02160            FDB    $0A0F
02170            FDB    $1913
02180            FDB    $1409
```

Listing continued

✓ See List of Advertisers on page 99

Listing continued

```
02190           FDB     $030B
02200           FDB     $2002
02210           FDB     $1514
02220           FDB     $140F
02230           FDB     $0E2E
02240 KEY       FDB     $0B05   CODES FOR
02250           FDB     $1902   'KEYBOARD OR
02260           FDB     $0F01   JOYSTICK? (K OR J)?'
02270           FDB     $1204
02280           FDB     $200F
02290           FDB     $1220
02300           FDB     $0A0F
02310           FDB     $1913
02320           FDB     $1409
02330           FDB     $030B
02340           FDB     $2028
02350           FDB     $0B20
02360           FDB     $0F12
02370           FDB     $200A
02380           FDB     $293F
02390 PRESSS    FDB     $1012   CODES FOR
02400           FDB     $0513   'PRESS SPACEBAR'
02410           FDB     $1320
02420           FDB     $1310
02430           FDB     $0103
02440           FDB     $0502
02450           FDB     $0112
02460 LOG       FDB     $AAAA   CODES FOR
02470           FDB     $AAAA   LOG
02480           FDB     $AAAA
02490           FDB     $AAAA
02500           FDB     $A000
02510           FDB     $0000
02520           FDB     $0000
02530           FDB     $02AA
02540           FDB     $8FFF
02550           FDB     $FFFF
02560           FDB     $C0FF
02570           FDB     $FCAA
02580           FDB     $8FFC
02590           FDB     $0FFF
02600           FDB     $FFFF
02610           FDB     $FCAA
02620           FDB     $3FFF
02630           FDB     $FFFF
02640           FDB     $FF00
02650           FDB     $0F2A
02660           FDB     $3FFF
02670           FDB     $C003
02680           FDB     $FFFF
02690           FDB     $FF2A
02700           FDB     $8C03
02710           FDB     $FFF5
02720           FDB     $FFFF
02730           FDB     $FCAA
02740           FDB     $8FFF
02750           FDB     $FFFF
02760           FDB     $FFC3
02770           FDB     $FCAA
02780           FDB     $A000
02790           FDB     $0000
02800           FDB     $0000
02810           FDB     $02AA
02820           FDB     $AAAA
02830           FDB     $AAAA
02840           FDB     $AAAA
02850           FDB     $AAAA
02860           ORG     $2000
02870 WAVES     FDB     $0D0E   WAVE TABLE
02880           FDB     $0F11   FOR MUSIC
02890           FDB     $1214   SUBROUTINE
02900           FDB     $1516
02910           FDB     $1819
02920           FDB     $1B1C
02930           FDB     $1D1F
02940           FDB     $2022
02950           FDB     $2326
02960           FDB     $2729
02970           FDB     $2A2B
02980           FDB     $2D2E
02990           FDB     $3031
03000           FDB     $3234
03010           FDB     $3537
03020           FDB     $3838
03030           FDB     $393B
03040           FDB     $3B3C
03050           FDB     $3C3E
03060           FDB     $3E3E
03070           FDB     $3F3F
03080           FDB     $3F3F
03090           FDB     $3F3F
03100           FDB     $3F3E
03110           FDB     $3E3E
03120           FDB     $3C3C
03130           FDB     $3B3B
03140           FDB     $3938
03150           FDB     $3737
03160           FDB     $3534
03170           FDB     $3231
03180           FDB     $302E
03190           FDB     $2E2D
03200           FDB     $2B2A
03210           FDB     $2927
03220           FDB     $2624
03230           FDB     $2322
03240           FDB     $201F
03250           FDB     $1F1D
03260           FDB     $1C1B
03270           FDB     $1B19
03280           FDB     $1918
03290           FDB     $1816
03300           FDB     $1615
03310           FDB     $1515
03320           FDB     $1515
03330           FDB     $1515
03340           FDB     $1515
03350           FDB     $1515
03360           FDB     $1515
03370           FDB     $1616
03380           FDB     $1618
03390           FDB     $1818
03400           FDB     $1919
03410           FDB     $1B1B
03420           FDB     $1C1C
03430           FDB     $1C1D
03440           FDB     $1D1F
03450           FDB     $1F1F
03460           FDB     $2020
03470           FDB     $2222
03480           FDB     $2222
03490           FDB     $2223
03500           FDB     $2323
03510           FDB     $2323
03520           FDB     $2323
03530           FDB     $2222
03540           FDB     $2222
03550           FDB     $2020
03560           FDB     $201F
03570           FDB     $1F1D
03580           FDB     $1D1C
03590           FDB     $1C1B
03600           FDB     $1919
03610           FDB     $1816
03620           FDB     $1515
03630           FDB     $1412
03640           FDB     $1211
03650           FDB     $0F0E
03660           FDB     $0E0D
03670           FDB     $0B0B
03680           FDB     $0A08
03690           FDB     $0807
03700           FDB     $0706
03710           FDB     $0604
03720           FDB     $0403
```

Listing continued

Listing continued

```
03730          FDB     $0303
03740          FDB     $0101
03750          FDB     $0101
03760          FDB     $0101
03770          FDB     $0000
03780          FDB     $0000
03790          FDB     $0001
03800          FDB     $0101
03810          FDB     $0101
03820          FDB     $0101
03830          FDB     $0303
03840          FDB     $0303
03850          FDB     $0304
03860          FDB     $0404
03870          FDB     $0404
03880          FDB     $0606
03890          FDB     $0606
03900          FDB     $0606
03910          FDB     $0606
03920          FDB     $0707
03930          FDB     $0707
03940          FDB     $0707
03950          FDB     $0706
03960          FDB     $0606
03970          FDB     $0606
03980          FDB     $0606
03990          FDB     $0606
04000          FDB     $0404
04010          FDB     $0404
04020          FDB     $0404
04030          FDB     $0404
04040          FDB     $0404
04050          FDB     $0303
04060          FDB     $0304
04070          FDB     $0404
04080          FDB     $0404
04090          FDB     $0406
04100          FDB     $0606
04110          FDB     $0707
04120          FDB     $0708
04130          FDB     $0A0A
04140          FDB     $0B0D
04150 TABLE    FDB     $0000   NOTE TABLE
04160          FDB     $026F   FOR MUSIC
04170          FDB     $0294   SUBROUTINE
04180          FDB     $02BB
04190          FDB     $02E4
04200          FDB     $0310
04210          FDB     $033F
04220          FDB     $0371
04230          FDB     $03A5
04240          FDB     $03DC
04250          FDB     $0417
04260          FDB     $0456
04270          FDB     $0498
04280          FDB     $04DD
04290          FDB     $0528
04300          FDB     $0576
04310          FDB     $05C9
04320          FDB     $0621
04330          FDB     $067F
04340          FDB     $06E2
04350          FDB     $074A
04360          FDB     $07B9
04370          FDB     $082F
04380          FDB     $08AC
04390          FDB     $0930
04400          FDB     $09BB
04410          FDB     $0A50
04420          FDB     $0AED
04430          FDB     $0B93
04440          FDB     $0C43
04450          FDB     $0CFE
04460          FDB     $0DC4
04470          FDB     $0E95
04480          FDB     $0F73
04490          FDB     $105F
04500          FDB     $1158
04510          FDB     $1260
04520          FDB     $1377
04530          FDB     $14A0
04540          FDB     $15DA
04550          FDB     $1726
04560          FDB     $1887
04570          FDB     $19FC
04580          FDB     $1B88
04590          FDB     $1D2B
04600          FDB     $1EE7
04610          FDB     $20BE
04620          FDB     $22B0
04630          FDB     $24C0
04640          FDB     $26EF
04650          FDB     $2940
04660          FDB     $2BB4
04670          FDB     $2E4D
04680          FDB     $310E
04690          FDB     $33F9
04700          FDB     $3710
04710          FDB     $3A57
04720          FDB     $3DCF
04730          FDB     $417C
04740          FDB     $4560
04750          FDB     $4981
04760          FDB     $4DDF
04770          FDB     $5281
04780          FDB     $5769
04790          FDB     $5C9B
04800          FDB     $621D
04810 RFROG    FDB     $FFFF   CODES FOR
04820          FDB     $75DF   FROGS ON
04830          FDB     $D57F   RED BACKGROUNDS
04840          FDB     $F5FF
04850          FDB     $F5FF
04860          FDB     $F5FF
04870          FDB     $F5FF
04880          FDB     $DF7F
04890          FDB     $7FDF
04900          FDB     $FFFF
04910          FDB     $FFFF
04920          FDB     $75DF
04930          FDB     $D57F
04940          FDB     $F5FF
04950          FDB     $F5FF
04960          FDB     $F5FF
04970          FDB     $F5FF
04980          FDB     $DF7F
04990          FDB     $F5FF
05000          FDB     $FFFF
05010          FDB     $FFFF
05020          FDB     $7FDF
05030          FDB     $DF7F
05040          FDB     $F5FF
05050          FDB     $F5FF
05060          FDB     $F5FF
05070          FDB     $F5FF
05080          FDB     $D57F
05090          FDB     $75DF
05100          FDB     $FFFF
05110          FDB     $FFFF
05120          FDB     $F5FF
05130          FDB     $DF7F
05140          FDB     $F5FF
05150          FDB     $F5FF
05160          FDB     $F5FF
05170          FDB     $F5FF
05180          FDB     $D57F
05190          FDB     $75DF
05200          FDB     $FFFF
05210          FDB     $FFFF
05220          FDB     $7FDF
05230          FDB     $DF7F
05240          FDB     $F57F
05250          FDB     $F57F
```

Listing continued

Listing continued

```
05260            FDB     $F57F
05270            FDB     $F57F
05280            FDB     $DF7F
05290            FDB     $7FDF
05300            FDB     $FFFF
05310            FDB     $FFFF
05320            FDB     $FFDF
05330            FDB     $DF7F
05340            FDB     $757F
05350            FDB     $757F
05360            FDB     $757F
05370            FDB     $757F
05380            FDB     $DF7F
05390            FDB     $FFDF
05400            FDB     $FFFF
05410            FDB     $FFFF
05420            FDB     $7FDF
05430            FDB     $DF7F
05440            FDB     $D5FF
05450            FDB     $D5FF
05460            FDB     $D5FF
05470            FDB     $D5FF
05480            FDB     $DF7F
05490            FDB     $7FDF
05500            FDB     $FFFF
05510            FDB     $FFFF
05520            FDB     $7FFF
05530            FDB     $DF7F
05540            FDB     $D5DF
05550            FDB     $D5DF
05560            FDB     $D5DF
05570            FDB     $D5DF
05580            FDB     $DF7F
05590            FDB     $7FFF
05600            FDB     $FFFF
05610 WFROG      FDB     $0000   CODES FOR
05620            FDB     $4510   FROGS ON
05630            FDB     $1540   WHITE BACKGROUNDS
05640            FDB     $0500
05650            FDB     $0500
05660            FDB     $0500
05670            FDB     $0500
05680            FDB     $1040
05690            FDB     $4010
05700            FDB     $0000
05710            FDB     $0000
05720            FDB     $4510
05730            FDB     $1540
05740            FDB     $0500
05750            FDB     $0500
05760            FDB     $0500
05770            FDB     $0500
05780            FDB     $1040
05790            FDB     $0500
05800            FDB     $0000
05810            FDB     $0000
05820            FDB     $4010
05830            FDB     $1040
05840            FDB     $0500
05850            FDB     $0500
05860            FDB     $0500
05870            FDB     $0500
05880            FDB     $1540
05890            FDB     $4510
05900            FDB     $0000
05910            FDB     $0000
05920            FDB     $0500
05930            FDB     $1040
05940            FDB     $0500
05950            FDB     $0500
05960            FDB     $0500
05970            FDB     $0500
05980            FDB     $1540
05990            FDB     $4510
06000            FDB     $0000
06010            FDB     $0000
```

Listing continued

Listing continued

```
06020          FDB     $4010
06030          FDB     $1040
06040          FDB     $0540
06050          FDB     $0540
06060          FDB     $0540
06070          FDB     $0540
06080          FDB     $1040
06090          FDB     $4010
06100          FDB     $0000
06110          FDB     $0000
06120          FDB     $0010
06130          FDB     $1040
06140          FDB     $4540
06150          FDB     $4540
06160          FDB     $4540
06170          FDB     $4540
06180          FDB     $1040
06190          FDB     $0010
06200          FDB     $0000
06210          FDB     $0000
06220          FDB     $4010
06230          FDB     $1040
06240          FDB     $1500
06250          FDB     $1500
06260          FDB     $1500
06270          FDB     $1500
06280          FDB     $1040
06290          FDB     $4010
06300          FDB     $0000
06310          FDB     $0000
06320          FDB     $4000
06330          FDB     $1040
06340          FDB     $1510
06350          FDB     $1510
06360          FDB     $1510
06370          FDB     $1510
06380          FDB     $1040
06390          FDB     $4000
06400          FDB     $0000
06410 BFROG    FDB     $AAAA   CODES FOR
06420          FDB     $659A   FROGS ON
06430          FDB     $956A   BLUE BACKGROUNDS
06440          FDB     $A5AA
06450          FDB     $A5AA
06460          FDB     $A5AA
06470          FDB     $A5AA
06480          FDB     $9A6A
06490          FDB     $6A9A
06500          FDB     $AAAA
06510          FDB     $AAAA
06520          FDB     $659A
06530          FDB     $956A
06540          FDB     $A5AA
06550          FDB     $A5AA
06560          FDB     $A5AA
06570          FDB     $A5AA
06580          FDB     $9A6A
06590          FDB     $A5AA
06600          FDB     $AAAA
06610          FDB     $AAAA
06620          FDB     $6A9A
06630          FDB     $9A6A
06640          FDB     $A5AA
06650          FDB     $A5AA
06660          FDB     $A5AA
06670          FDB     $A5AA
06680          FDB     $956A
06690          FDB     $659A
06700          FDB     $AAAA
06710          FDB     $AAAA
06720          FDB     $A5AA
06730          FDB     $9A6A
06740          FDB     $A5AA
06750          FDB     $A5AA
06760          FDB     $A5AA
06770          FDB     $A5AA
```

Listing continued

⌐ See List of Advertisers on page 99

Listing continued

```
06780            FDB     $956A
06790            FDB     $659A
06800            FDB     $AAAA
06810            FDB     $AAAA
06820            FDB     $6A9A
06830            FDB     $9A6A
06840            FDB     $A56A
06850            FDB     $A56A
06860            FDB     $A56A
06870            FDB     $A56A
06880            FDB     $9A6A
06890            FDB     $6A9A
06900            FDB     $AAAA
06910            FDB     $AAAA
06920            FDB     $AA9A
06930            FDB     $9A6A
06940            FDB     $656A
06950            FDB     $656A
06960            FDB     $656A
06970            FDB     $656A
06980            FDB     $9A6A
06990            FDB     $AA9A
07000            FDB     $AAAA
07010            FDB     $AAAA
07020            FDB     $6A9A
07030            FDB     $9A6A
07040            FDB     $95AA
07050            FDB     $95AA
07060            FDB     $95AA
07070            FDB     $95AA
07080            FDB     $9A6A
07090            FDB     $6A9A
07100            FDB     $AAAA
07110            FDB     $AAAA
07120            FDB     $6AAA
07130            FDB     $9A6A
07140            FDB     $959A
07150            FDB     $959A
07160            FDB     $959A
07170            FDB     $959A
07180            FDB     $9A6A
07190            FDB     $6AAA
07200            FDB     $AAAA
07210 WSKULL     FDB     $0FC0   CODES FOR
07220            FDB     $0300   SKULL ON
07230            FDB     $0FC0   WHITE BACKGROUND
07240            FDB     $0CC0
07250            FDB     $0CC0
07260            FDB     $0300
07270            FDB     $3330
07280            FDB     $0CC0
07290            FDB     $0300
07300            FDB     $0CC0
07310            FDB     $3030
07320 BSKULL     FDB     $AFEA   CODES FOR
07330            FDB     $ABAA   SKULL ON
07340            FDB     $AFEA   BLUE BACKGROUND
07350            FDB     $AEEA
07360            FDB     $AEEA
07370            FDB     $ABAA
07380            FDB     $BBBA
07390            FDB     $AEEA
07400            FDB     $ABAA
07410            FDB     $AEEA
07420            FDB     $BABA
07430 BFROGF     FDB     $AAAA   CODES FOR
07440            FDB     $A659   FROG IN
07450            FDB     $A956   SCORE PORT
07460            FDB     $AA5A
07470            FDB     $AA5A
07480            FDB     $AA5A
07490            FDB     $AA5A
07500            FDB     $A9A6
07510            FDB     $A6A9
07520            FDB     $AAAA
07530 WTURT      FDB     $AAAA   CODES FOR
07540            FDB     $AAAA   SET OF
07550            FDB     $AAAA   UNDERWATER
07560            FDB     $AAAA   TURTLES
07570            FDB     $AAAA
07580            FDB     $AAAA
07590            FDB     $AAAA
07600            FDB     $AAAA
07610            FDB     $AAAA
07620            FDB     $EAAA
07630            FDB     $AEAA
07640            FDB     $AAEA
07650            FDB     $AAAB
07660            FDB     $FAAA
07670            FDB     $BFAA
07680            FDB     $ABFA
07690            FDB     $AAAF
07700            FDB     $FEAA
07710            FDB     $FFEA
07720            FDB     $AFFE
07730            FDB     $AAAF
07740            FDB     $FEAA
07750            FDB     $FFEA
07760            FDB     $AFFE
07770            FDB     $AAAB
07780            FDB     $FAAA
07790            FDB     $BFAA
07800            FDB     $ABFA
07810            FDB     $AAAA
07820            FDB     $EAAA
07830            FDB     $AEAA
07840            FDB     $AAEA
07850            FDB     $AAAA
07860            FDB     $AAAA
07870            FDB     $AAAA
07880            FDB     $AAAA
07890            FDB     $AAAA
07900            FDB     $AAAA
07910            FDB     $AAAA
07920            FDB     $AAAA
07930 TURTLE     FDB     $AAAA   CODES FOR
07940            FDB     $AAAA   SET OF
07950            FDB     $AAAA   TURTLES
07960            FDB     $AAAA
07970            FDB     $AAAA
07980            FDB     $EAAA
07990            FDB     $AEAA
08000            FDB     $AAEA
08010            FDB     $AAAB
08020            FDB     $FAAA
08030            FDB     $BFAA
08040            FDB     $ABFA
08050            FDB     $AAAC
08060            FDB     $0EAA
08070            FDB     $C0EA
08080            FDB     $AC0E
08090            FDB     $AABC
08100            FDB     $0FAB
08110            FDB     $C0FA
08120            FDB     $BC0F
08130            FDB     $AABC
08140            FDB     $0FAB
08150            FDB     $C0FA
08160            FDB     $BC0F
08170            FDB     $AAAC
08180            FDB     $0EAA
08190            FDB     $C0EA
08200            FDB     $AC0E
08210            FDB     $AAAB
08220            FDB     $FAAA
08230            FDB     $BFAA
08240            FDB     $ABFA
08250            FDB     $AAAA
08260            FDB     $EAAA
08270            FDB     $AEAA
08280            FDB     $AAEA
08290            FDB     $AAAA
08300            FDB     $AAAA
```

Listing continued

Listing continued

```
08310           FDB     $AAAA
08320           FDB     $AAAA
08330 CAR1      FDB     $0000   CODES FOR
08340           FDB     $0000   FIRST CAR
08350           FDB     $0FC3
08360           FDB     $F0FC
08370           FDB     $0555
08380           FDB     $5554
08390           FDB     $0555
08400           FDB     $55A4
08410           FDB     $0555
08420           FDB     $55A4
08430           FDB     $0555
08440           FDB     $55A4
08450           FDB     $0555
08460           FDB     $55A4
08470           FDB     $0555
08480           FDB     $5554
08490           FDB     $0FC3
08500           FDB     $F0FC
08510           FDB     $0000
08520           FDB     $0000
08530 CAR2      FDB     $0000   CODES FOR
08540           FDB     $0000   SECOND CAR
08550           FDB     $0541
08560           FDB     $5054
08570           FDB     $0FFF
08580           FDB     $FFFC
08590           FDB     $0FAF
08600           FDB     $FFFC
08610           FDB     $0FAF
08620           FDB     $FFFC
08630           FDB     $0FAF
08640           FDB     $FFFC
08650           FDB     $0FAF
08660           FDB     $FFFC
08670           FDB     $0FFF
08680           FDB     $FFFC
08690           FDB     $0541
08700           FDB     $5054
08710           FDB     $0000
08720           FDB     $0000
08730 CAR3      FDB     $0000   CODES FOR
08740           FDB     $0000   THIRD CAR
08750           FDB     $0541
08760           FDB     $5054
08770           FDB     $0AAA
08780           FDB     $AAA8
08790           FDB     $0AAA
08800           FDB     $AAF8
08810           FDB     $0AAA
08820           FDB     $AAF8
08830           FDB     $0AAA
08840           FDB     $AAF8
08850           FDB     $0AAA
08860           FDB     $AAF8
08870           FDB     $0AAA
08880           FDB     $AAA8
08890           FDB     $0541
08900           FDB     $5054
08910           FDB     $0000
08920           FDB     $0000
08930 CAR4      FDB     $0000   CODES FOR
08940           FDB     $0000   FOURTH CAR
08950           FDB     $0FC3
08960           FDB     $F0FC
08970           FDB     $0AAA
08980           FDB     $AAA8
08990           FDB     $0A0A
09000           FDB     $AAA8
09010           FDB     $0A0A
09020           FDB     $AAA8
09030           FDB     $0A0A
09040           FDB     $AAA8
09050           FDB     $0A0A
09060           FDB     $AAA8
09070           FDB     $0AAA
09080           FDB     $AAA8
09090           FDB     $0FC3
09100           FDB     $F0FC
09110           FDB     $0000
09120           FDB     $0000
09130 CAR5      FDB     $0000   CODES FOR
09140           FDB     $0000   FIFTH CAR
09150           FDB     $0A82
09160           FDB     $A0A8
09170           FDB     $0555
09180           FDB     $5554
09190           FDB     $0555
09200           FDB     $55F4
09210           FDB     $0555
09220           FDB     $55F4
09230           FDB     $0555
09240           FDB     $55F4
09250           FDB     $0555
09260           FDB     $55F4
09270           FDB     $0555
09280           FDB     $5554
09290           FDB     $0A82
09300           FDB     $A0A8
09310           FDB     $0000
09320           FDB     $0000
09330 CAR6      FDB     $0000   CODES FOR
09340           FDB     $0000   SIXTH CAR
09350           FDB     $0A82
09360           FDB     $A0A8
09370           FDB     $0FFF
09380           FDB     $FFFC
09390           FDB     $0F0F
09400           FDB     $FFFC
09410           FDB     $0F0F
09420           FDB     $FFFC
09430           FDB     $0F0F
09440           FDB     $FFFC
09450           FDB     $0F0F
09460           FDB     $FFFC
09470           FDB     $0FFF
09480           FDB     $FFFC
09490           FDB     $0A82
09500           FDB     $A0A8
09510           FDB     $0000
09520           FDB     $0000
09530 NAME      FDB     $0312   CODES FOR
09540           FDB     $0F01   'CROAKER'
09550           FDB     $0B05
09560           FCB     $12
09570 CRED      FDB     $0219   CODES FOR
09580           FDB     $200D   'BY: MIKE MEEHAN'
09590           FDB     $090B
09600           FDB     $0520
09610           FDB     $0D05
09620           FDB     $0508
09630           FDB     $010E
09640           FCB     $20
09650 COP       FDB     $030F   CODES FOR
09660           FDB     $1019   'COPYRIGHT 1983'
09670           FDB     $1209
09680           FDB     $0708
09690           FDB     $1420
09700           FDB     $3139
09710           FDB     $3833
09720 COMP      FDB     $030F   CODES FOR
09730           FDB     $0C0F   'COLOR HORIZONS
09740           FDB     $1220   SOFTWARE'
09750           FDB     $080F
09760           FDB     $1209
09770           FDB     $1A0F
09780           FDB     $0E13
09790           FDB     $2013
09800           FDB     $0F06
09810           FDB     $1417
09820           FDB     $0112
09830           FCB     $05
09840           END
```

END

Macrotron**
Proudly Introduces Our New *"Premium" Keyboard—*
The Best For Your Color Computer

"Have Josie ship yours today!"

The Best Keyboard

All the features of our popular PROFESSIONAL KEYBOARD:
No gluing, soldering, or cutting—plugs right in.
High quality construction assures years of trouble-free operation.

PLUS

Attractive low profile
Extended Radio Shack layout
Silk-Smooth feel

The Best Software

Our Versakey Software enhances the keyboard's utility:

* Auto-repeat, n-key rollover and type-ahead
* F-1 becomes DEFINE, F-4 becomes CTRL
* May define up to 128 keys (including SHIFT, CTRL, and SHIFT-CTRL combinations) as strings of up to 80 characters each.
* Supplied on cassette, may be copied to disk.

The Best Manual

* Very complete documentation (including plenty of figures to illustrate the keyboard's installation and versatility).

The Best Prices

The "Premium" Keyboard (including software)	$79.95
The "Professional" Keyboard (including software)	$59.95
The "Versakey Software"	$ 9.95

Please specify your computer's PC board type if known. Otherwise, specify the complete catalog number and serial number.

**Micronix Systems is a subdivision of Macrotron Systems Corporation.

MACROTRON SYSTEMS CORP. 203

8147 Delmar Blvd.
St. Louis, MO 63130
(314) 721-3356
Telex 704523 Answer Back-MACANITA STL UD

Terms: Prepaid check or money order, Mastercard or Visa.
Shipping Charges: U.S. $3.00, Canada $6.00, COD $5.00 (No COD's to Canada).

Python

As you snake through the maze consuming dots, be careful of the obstacles that can cut you short.

GAME
BY ANDREW SIDDELEY

In Python, you must guide a snake (python) through a maze, eating all the dots in the process. Every time you hit a wall or stop, the tail keeps moving straight ahead. If you are not careful, the tail will catch up to the head, and the game will end. You also must avoid hesitating at intersections or getting tangled up in your tail.

Continued on p. 64

A$(1)-A$(15)	Contain the maze in string form.
SC	Score.
OB	Objective. When the score equals this, it means that all the dots on the screen have been eaten.
SN	Screen counter. This keeps count of the number of screens that the player has cleared.
PC	Python color. 1 = red, 4 = orange, 2 = white, 5 = magenta, 3 = yellow, 6 = cyan, P$(PC) = a colored block
P	Python position. This is the position of the head of the python.
LP	Length of python. This is the size of the stack, or the number of positions the head is from the tail.
TR(LP)	The stack. This is an array LP elements long, which contains the positions of the python's body.
ST	The start pointer. This points to the head of the python. The new python position (P) is put into the array at TR(ST).
EN	The end pointer. This points to the tail of the python. This position is reset by printing TR(EN)," ";.
DL	Difficulty. This variable is for the delay loop in line 330. It controls the tempo of the game.
PE	PEEK value. Line 400 PEEKs into the next position the python will occupy and returns the ASCII code of that position.
DV(1)-DV(5)	Direction vector. DV(1) = −32↑ DV(2) = +32↓ DV(3) = −1← DV(4) = +1→ DV(5) = 0 (stationary)

Table 1. Main Variables

If you eat all the dots, you receive bonus points based on how much of the python is left. A little melody plays, and another maze filled with dots appears. After each screen, the python changes color and grows a little shorter, and the tempo of the game increases.

You control the python with the following keys:

K—up
<—down
L—left
+—right

Eliminating line 510 of the program will speed up the game considerably, but you will see your score only during the interlude.

If a BS (bad subscript) error occurs, change line 50 to DIM C(20),L(20), TR(150),A$(20). Otherwise, you should be all set to go. ■

Address correspondence to Andrew Siddeley, 101 Catalina Drive, Scarborough, Ontario M1M 1K7, Canada.

System Requirements

16K RAM
Extended Color Basic

Program Listing. Python

```
30 CLEAR1000
35 'title*page
40 CLS:PRINT@42,STRING$(11,"*"):
   PRINT@74,"P Y T H O N":
   PRINT@106,STRING$(11,"*"):
   PRINT@258,"CREATED BY: ANDREW
 SIDDELEY":PRINT"  [FRI/10/SEPT/
82]":PRINT
45 'initialization
50 DIML(20),C(20),TR(125),A$(20)
60 DV(1)=-32 : DV(2)=32 :
   DV(3)=-1  : DV(4)=1  :
   DV(5)=0
70 P$(1)=CHR$(191) : P$(2)=CHR$
   (207) : P$(3)=CHR$(159) :
   P$(4)=CHR$(255) : P$(5)=CHR$
   (239) : P$(6)=CHR$(223)
80 'compose*maze
   [MAZE IS KEPT IN STING FORM]
90 FORT=1TO15
100 A=A+1
110 READ L(A),C(A)
120 IFL(A)=99THEN150
130 A$(T)=A$(T)+STRING$(L(A),
    C(A))
140 GOTO100
150 A=A-1:IFA<1THENA=0:NEXTT:
    GOTO180
160 A$(T)=A$(T)+STRING$(L(A),
    C(A))
170 GOTO150
180 'input*level*of*play
190 INPUT" ENTER THE LEVEL OF DI
FFICULTY   [HARD(1)-EASY(8)] ";
    DF:IFDF<1 OR DF>8 THENDF=4
200 DL=DF
210 INPUT" ENTER THE LENGTH OF T
HE PYTHON  [10-150] ";LP:IFLP<10
    OR LP>150 THEN LP=100
215 'menu
220 CLS:PRINT@74,"P Y T H O N":
    PRINT
230 PRINT" D)EMO":
    PRINT" I)NSTRUCTIONS":
    PRINT" C)HANGE LEVEL":
    PRINT" S)TART"
240 K$=INKEY$
250 IF K$="D" THEN 280
260 IF K$="S" THEN 690
265 IF K$="I" THEN 1000
267 IF K$="C" THEN 180
270 GOTO 240
275 'draw*maze
280 'CLS:FORT=1TO15:PRINTA$(T);:
    NEXTT:GOSUB770
290 P=230:DI=4:ST=0:EN=1
300 FORA=1TOLP:TR(A)=P:NEXTA
310 IF K$="D" THEN 810
320 'main*program*cycle
330 FOR C=1 TO DL
340 C$=INKEY$
350 IF C$=";" THEN DI=4
360 IF C$="L" THEN DI=3
370 IF C$="K" THEN DI=1
380 IF C$="," THEN DI=2
390 NEXTC
400 PE=PEEK(1024+P+DV(DI))
410 IF PE=110 THEN SC=SC+1:
    IF SC=OB THEN PRINT@P+DV
    (DI),P$(PC);:GOTO540
420 IF PE>128 THEN F=1:DI=5
    ELSE F=0
430 P=P+DV(DI)
440 PRINT@P,P$(PC);
450 IF F=1 THEN 480
460 ST=ST+1:IF ST>LP THEN ST=1
470 TR(ST)=P
480 EN=EN+1:IF EN>LP THEN EN=1
490 PRINT@TR(EN)," ";
500 IF ST=EN THEN 910
510 'GOSUB 780
520 GOTO320
530 'end*of*main*cycle
531 '
535 'calculate*how*much*of*the*
     python*is*left*and*give*
     bonus*points
540 EN=EN+1:IFEN>LP THEN EN=1
550 PLAY"T100O2CDABGFAF":PRINT@
    TR(EN)," ";
560 SC=SC+10:GOSUB780
```

Listing continued

Listing continued

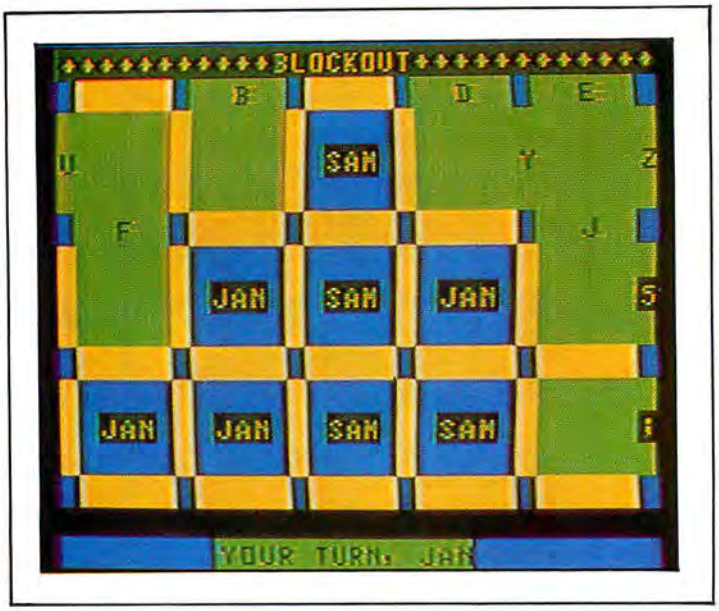
```
570 IF EN<>ST THEN 540
575 'compose*melody*with*random
    tempo*and*octave
580 O=RND(4)+1:T=RND(4)+2
590 T$(0)="O"+STR$(O)+"T"+STR$
    (T)
600 T$(1)="L16CP16CP16L8EP8"
610 T$(2)=T$(1)
620 T$(3)="L4CP8L16CP32L4DP8L16F
    P32FP8L16FP32EP8L8DP16CEP4"
630 T$(4)="L8CP64CP32DP64DP32FP6
    4FP32L4EP4"
640 'T$(5)="L8DP16CP16O-L8BABO+CP
    16L16CP32CP32CP2"
645 'play*the*melody
650 FORT=0TO5:PLAYT$(T):NEXTT
655 'interlude
660 CLS:PRINT:PRINTTAB(3);
   "YOU FINISHED THIS SCREEN !!"

665 'increase*difficulty[DF]
    and*shorten*the*python[LP]
670 LP=LP-20:IF LP<10 THEN LP=10
680 DF=DF-.5:DL=INT(DF): IF DL<1
    THEN DF=1:DL=1
690 OB=SC+229:SN=SN+1
700 PC=PC+1:IF PC>6 THEN PC=1
710 PLAY"T4L4CDEFGAB":PRINT:
    PRINT:PRINT" score:";SC:
    PRINT:PRINT" length of pytho
n:"LP:PRINT:PRINT" screens:  ";
720 FORT=1 TO SN: PLAY"T200DAC":
    PRINTP$(PC);:PRINT" ";:FORTI
    =1 TO 100: NEXTTI,T
730 'PLAY"T4L4BAGFEDC"
740 FORT=1TO1000:NEXTT
750 GOTO 280
760 'print*the*score*on*screen
    subroutine
770 PRINT@480,"SCORE:";
780 PRINT@488,"";:PRINT USING
    "##,###";SC;
790 RETURN
800 'demo*program
810 PRINT@480," ** PRESS ANY KEY
 TO RETURN **";
820 IF PEEK(1024+P+DV(DI))>128
    THEN DI=RND(4): GOTO820
830 P=P+DV(DI)
840 PRINT@P,P$(RND(6));
850 ST=ST+1:IF ST>10 THEN ST=1
860 EN=EN+1:IF EN>10 THEN EN=1
870 TR(ST)=P
880 PRINT@TR(EN)," ";
890 IF INKEY$<>""THEN220
900 GOTO820
905 'end*of*demo*program*cycle
906 '
907 'game*over*mesage
910 'FORT=1TO90STEP2:SCREEN0,0:
    FORT1=1TOT:NEXTT1:SCREEN0,1:
    PLAY"T200O1DAC":NEXTT:SCREEN
    0,0:PLAY"T4O3L2AP16EP16L1O1E
    #"
920 PRINT@231,"TOO BAD, GAME OVE
R";
930 FORT=1TO1500:NEXTT
940 PRINT@224,"DO YOU WANT TO PL
AY AGAIN  [Y/N]";
950 K$=INKEY$
960 IFK$="Y" THEN CLS:SC=0:PC=0:
    SN=0:GOTO215
970 IFK$="N" THEN CLS:END
980 GOTO950
999 'instructions
1000 CLS:PRINT
1010 PRINT"      THE OBJECT OF T
HIS GAME    IS TO GUIDE THE PYTH
ON THROUGH  THE MAZE AND EAT UP
ALL THE   "
1020 PRINT" DOTS BEFORE THE TAIL
 CATCHES    UP WITH THE HEAD, AN
D BLOWS     THE PYTHON TO TINY B
ITS."
1030 PRINT:PRINT"      THE CONTR
OLS ARE :"
1035 'PRINT
1040 PRINTTAB(12);"K  UP":
     PRINTTAB(12);"<  DOWN":
     PRINTTAB(12);"L  LEFT":
     PRINTTAB(12);"+  RIGHT"
1050 PRINT:PRINT"   ** PRESS ENT
ER TO RETURN **";
1060 IF INKEY$=CHR$(13) THEN 220
     ELSE 1060
1090 'data*for*the*maze
2000 DATA 16,175,99,99
2010 DATA 1,175,12,46,1,175,2,46
     ,99,99
2020 DATA 1,175,1,46,7,175,1,46,
     2,175,1,46,1,175,1,46,1,175
     ,99,99
2030 DATA 1,175,7,46,1,175,1,46,
     2,175,1,46,1,175,1,46,1,175
     ,99,99
2040 DATA 1,175,1,46,2,175,1,46,
     2,175,1,46,1,175,1,46,2,175
     ,1,46,1,175,1,46,1,175,99,
     99
2050 'DATA 1,175,1,46,2,175,1,46,
2,175,1,46,1,175,6,46,1,175,99,9
9
2060 DATA 1,175,4,46,2,175,1,46,
1,175,1,46,6,175,99,99
2070 DATA 4,175,12,46,99,99
2080 DATA 1,175,4,46,2,175,1,46,
1,175,1,46,2,175,1,46,3,175,99,9
9
2090 DATA 1,175,1,46,2,175,1,46,
2,175,1,46,1,175,6,46,1,175,99,9
9
2100 DATA 1,175,1,46,2,175,1,46,
2,175,1,46,1,175,1,46,2,175,1,46
,1,175,1,46,1,175,99,99
2110 DATA 1,175,14,46,1,175,99,9
9
2120 DATA 1,175,1,46,5,175,1,46,
4,175,1,46,1,175,1,46,1,175,99,9
9
2130 DATA 1,175,12,46,1,175,2,46
,99,99
2140 DATA 16,175,99,99
```

END

✓ See List of Advertisers on page 99

BY KEN KNUDSON



COLOR BLOCKOUT

This 2,500-byte game packs a lot of fun into a 4K CoCo. The game's object is simple—two players compete against one another to connect points on a grid to form squares.

System Requirements

4K RAM
Color Basic
or
MC-10

4K CoCo owners rejoice! Blockout proves that a 2,500-byte game can pack a big entertainment punch.

The grid is composed of blue dots separated by letters and numbers. Whoever draws the fourth line to form a square wins it. The computer places their initials, or three letters, on the square. The maximum number of squares possible is 15 (due to the limitations of the print-graphics screen). Therefore, the player with eight or more squares wins.

While Blockout's concept is simple, it takes strategy to win. I have provided both Color Basic and MC-10 game versions. ■

Address correspondence to Ken Knudson, 2001 Vahlen, Madison, WI 53704.

Program Listing 1. Color Basic Blockout

```
10 CLEAR50
20 CLEAR:N$=CHR$(128):CLS
30 PRINT@129,"TYPE IN YOUR INITI
ALS OR NAME";:PRINT@160,"(AT LEA
ST THREE LETTERS, PLEASE)";:PRIN
T@11,"blockout";:PRINT:PRINT@195
,"PRESS ENTER AFTER EACH NAME"
40 PRINT@264,"NAME";:INPUTA$:IFL
EN(A$)<3 THENA$=A$+"ZZ
50 PRINT@328,"NAME";:INPUTB$:IFL
EN(B$)<3THENB$=B$+"ZZ
60 CLS:FORX=1024TO1055:POKEX,42:
NEXT:PRINT@11,"blockout";
70 FORX=1056TO1440STEP128:FORY=X
 TOX+30STEP6:POKEY,175:NEXTY,X
80 Z=65:FORX=1059TO1443STEP128:F
ORY=X TOX+24STEP6:POKEY,Z:Z=Z+1:
NEXTY,X
90 Z=85:FORX=1120TO1150STEP6:POK
EX,Z:Z=Z+1:NEXTX
100 Z=48:FORX=1248TO1376STEP128:
FORY=X TOX+30 STEP6:POKEY,Z:Z=Z+
1:NEXTY,X
110 FORX=1055TO1535STEP32:POKEX,
128:NEXT
120 FORX=1472TO1503:POKEX,128:NE
XT:GOSUB160
130 P=RND(2):IFP=2THEN150
140 GOSUB160:PRINT@488,"YOUR TUR
N, ";A$;:C$=A$:GOTO170
150 GOSUB160:PRINT@488,"YOUR TUR
N, ";B$;:C$=B$:GOTO170
160 PP=0:FORX=1504TO1534:POKEX,1
75:NEXT:RETURN
170 I$=INKEY$:IFI$=""THEN170
180 SOUND220,1:V=ASC(I$):IFV<480
RV>90 THEN SOUND100,5:GOTO170
190 IFV>59ANDV<65THENSOUND100,5:
GOTO170
200 GOSUB480:IFF=1THEN220
210 GOSUB510:IFF<>1THENSOUND100,
5:GOTO170
220 SOUND200,1:IFC/2=INT(C/2)THE
NFORX=C-32TOC+32STEP32:POKEX,159
:NEXTX:GOTO240
```

Listing continued

Listing continued

```
230 FORX=C-2TOC+2:POKEX,159:NEXT
240 FORX=170TO230STEP15:SOUNDX,1
:NEXT
250 IFC/2=INT(C/2)THENGOSUB310:G
OTO270
260 GOSUB350:GOTO270
270 IFA+B=15THENFORX=10TO250STEP
10:SOUNDX,1:NEXT:GOTO420
280 IFPP=1ANDP=1THEN140ELSEIFPP=
1ANDP=2THEN150
290 P=P+1:IFP=3THENP=1
300 IFP=1THEN140ELSE150
310 IFPEEK(C-6)=159ANDPEEK(C-67)
=159ANDPEEK(C+61)=159THEN320ELSE
330
320 G=C-37:GOSUB540:G=C-4:GOSUB5
50
330 IFPEEK(C+6)=159ANDPEEK(C-61)
=159ANDPEEK(C+67)=159THEN340ELSE
RETURN
340 G=C-31:GOSUB540:G=C+2:GOSUB5
50:RETURN
350 IFPEEK(C-128)=159ANDPEEK(C-6
7)=159ANDPEEK(C-61)=159THEN360EL
SE370
360 G=C-98:GOSUB540:G=C-65:GOSUB
550
370 IFPEEK(C+128)=159ANDPEEK(C+6
1)=159ANDPEEK(C+67)=159THEN380EL
SERETURN
380 G=C+30:GOSUB540:G=C+63:GOSUB
550:RETURN
390 IFP=1THENA=A+1ELSEB=B+1
400 FORX=1TO5:FORY=180TO220STEP2
0:SOUNDY,1:NEXTY,X
410 RETURN
420 CLS3:IFA>B THENC$=A$:GOSUB44
0:GOTO460
430 C$=B$:GOSUB440:GOTO460
440 H$=C$+" "+"WINS!!!!!":J$="**
"+C$+"**"
450 FORX=1TO7:PRINTTAB(9)H$:PRIN
TTAB(12)J$:NEXTX:RETURN
460 PRINT@482,"TO PLAY AGAIN, PR
ESS ANY KEY";
470 IFINKEY$=""THEN470ELSE20
480 FORD=1059TO1443STEP128:FORE=
D TOD+24STEP6
490 F=0:IFPEEK(E)=V THENC=E:D=14
43:E=1467:F=1:NEXTE,D:RETURN
500 NEXTE,D:RETURN
510 FORD=1120TO1376STEP128:FORE=
D TOD+30STEP6
520 F=0:IFPEEK(E)=V THENC=E:D=13
76:E=1406:F=1:NEXTE,D:RETURN
530 NEXTE,D:RETURN
540 FORX=G TOG+64STEP32:FORY=X T
OX+4:POKEY,175:SOUNDRND(255),1:N
EXTY,X:RETURN
550 POKEG,ASC(LEFT$(C$,1))-64:SO
UND212,1:POKEG+1,ASC(MID$(C$,2,1
))-64:SOUND220,1:POKEG+2,ASC(MID
$(C$,3,1))-64:SOUND230,1:GOSUB39
0:PP=1:RETURN
```

END

Program Listing 2. MC-10 Blockout

```
100 REM BLOCKOUT FOR MC10--BY KE
N KNUDSON
110 CLS:PRINT@129,"TYPE IN YOUR
INITIALS OR NAME";
120 PRINT@160,"(AT LEAST THREE L
ETTERS, PLEASE)";
130 PRINT@11,"blockout";
140 PRINT@195,"PRESS ENTER AFTER
 EACH NAME"
150 PRINT@264,"NAME";:INPUTA$:IF
LEN(A$)<3 THENA$=A$+"ZZ"
160 PRINT@328,"NAME";:INPUTB$:IF
LEN(B$)<3 THENB$=B$+"ZZ"
170 CLS:FORX=16384TO16415:POKEX,
42:NEXT:PRINT@11,"blockout";
180 FORX=16416TO16800STEP128:FOR
Y=X TOX+30STEP6:POKEY,175:NEXTY,
X
190 Z=65:FORX=16419TO16803STEP12
8:FORY=X TOX+24STEP6:POKEY,Z:Z=Z
+1:NEXTY,X
200 Z=85:FORX=16480TO16510STEP6:
POKEX,Z:Z=Z+1:NEXTX
210 Z=48:FORX=16608TO16736STEP12
8:FORY=X TOX+30STEP6:POKEY,Z:Z=Z
+1:NEXTY,X
220 FORX=16415TO16895STEP32:POKE
X,128:NEXT
230 FORX=16832TO16863:POKEX,128:
NEXT:GOSUB270
240 P=RND(2):IFP=2 THEN260
250 GOSUB270:PRINT@488,"YOUR TUR
N, ";A$;:C$=A$:GOTO280
260 GOSUB270:PRINT@488,"YOUR TUR
N, ";B$;:C$=B$:GOTO280
270 PP=0:FORX=16864TO16894:POKEX
,175:NEXT:RETURN
280 I$=INKEY$:IFI$=""THEN280
290 SOUND220,1:V=ASC(I$):IFV<480
RV>90 THENSOUND100,5:GOTO280
300 IFV>59ANDV<65THENSOUND100,5:
GOTO280
310 GOSUB670:IFF=1THEN330
320 GOSUB700:IFF<>1THENSOUND100,
5:GOTO280
330 SOUND200,1:IFC/2=INT(C/2)THE
NFORX=C-32TOC+32STEP32:POKEX,159
:NEXTX:GOTO350
340 FORX=C-2TOC+2:POKEX,159:NEXT
350 FORX=170TO230STEP15:SOUNDX,1
:NEXT
360 IFC/2=INT(C/2)THENGOSUB440:G
OTO380
370 GOSUB500:GOTO380
380 IFA+B=15THENFORX=10TO250STEP
10:SOUNDX,1:NEXT:GOTO600
390 IFPP=1ANDP=1THEN250
400 IFPP=1ANDP=2THEN260
410 P=P+1:IFP=3THENP=1
420 IFP=1THEN250
430 GOTO260
440 IFPEEK(C-6)=159ANDPEEK(C-67)
=159ANDPEEK(C+61)=159THEN460
450 GOTO470
460 G=C-37:GOSUB730:G=C-4:GOSUB7
40
470 IFPEEK(C+6)=159ANDPEEK(C-61)
=159ANDPEEK(C+67)=159THEN490
480 RETURN
490 G=C-31:GOSUB730:G=C+2:GOSUB7
40:RETURN
500 IFPEEK(C-128)=159ANDPEEK(C-6
7)=159ANDPEEK(C-61)=159THEN520
510 GOTO530
520 G=C-98:GOSUB730:G=C-65:GOSUB
740
530 IFPEEK(C+128)=159ANDPEEK(C+6
1)=159ANDPEEK(C+67)=159THEN550
540 RETURN
550 G=C+30:GOSUB730:G=C+63:GOSUB
740:RETURN
560 IFP=1THENA=A+1:GOTO580
570 B=B+1
580 FORX=1TO5:FORY=180TO220STEP2
0:SOUNDY,1:NEXTY,X
590 RETURN
600 CLS3:IFA>B THENC$=A$:GOSUB62
0:GOTO640
610 C$=B$:GOSUB620:GOTO640
620 H$=C$+" "+"WINS!!!!!":J$="**
"+C$+"**"
630 FORX=1TO7:PRINTTAB(9)H$:PRIN
TTAB(12)J$:NEXTX:RETURN
640 PRINT@482,"TO PLAY AGAIN, PR
ESS ANY KEY";
650 IFINKEY$=""THEN650
660 CLEAR:GOTO110
670 FORD=16419TO16803STEP128:FOR
E=D TOD+24STEP6
680 F=0:IFPEEK(E)=V THENC=E:D=16
803:E=16827:F=1:NEXTE,D:RETURN
690 NEXTE,D:RETURN
700 FORD=16480TO16736STEP128:FOR
E=D TOD+30STEP6
710 F=0:IFPEEK(E)=V THENC=E:D=16
736:E=16766:F=1:NEXTE,D:RETURN
720 NEXTE,D:RETURN
730 FORX=G TOG+64STEP32:FORY=X T
OX+4:POKEY,175:SOUNDRND(255),1:N
EXTY,X:RETURN
740 POKEG,ASC(LEFT$(C$,1))-64:SO
UND212,1
750 POKEG+1,ASC(MID$(C$,2,1))-64
:SOUND220,1
760 POKEG+2,ASC(MID$(C$,3,1))-64
:SOUND230,1
770 GOSUB560:PP=1:RETURN
```

END

↙ See List of Advertisers on page 99

BY ANNA M. REEVES

A COLLECTOR'S ITEM (SORT OF)

For just about every item that exists, you'll find someone who collects it. Some people collect golf tees, others have shelves of china cups, sea shells, shotguns, racks of coins, albums of stamps, and the list goes on. My son has a hat collection and likes to keep an alphabetical list ordered according to the source of each hat.

I've written a machine-language sort subroutine and merged it with a Basic program (Listing 2). It takes .35 seconds to sort 117 hats, and about 10 seconds to sort 1,000 strings.

Do you keep your bottle-cap collection in a bag? Your prize insects in a shoe box? Get organized.

You can alter this program to keep track of your own collection or to sort any one- or two-dimensional string array. (If we kept track of name, source, color, year, and manufacturer, I would have used DIM A$(630,4) to sort by any of the five columns (0–4).) Change the program to meet your needs by incorporating the lines in Table 1.

You can also add to your list, delete or change items, store and retrieve from tape, and make a printout. The A key in lines 280 and 340 allows you to abort a printout without hitting the break key and losing your information.

As in all machine-language subroutines, be sure to make a copy of your program before you run it. This is especially true here since line 90 deletes lines to give you more string space.

Line 20	Clears room for machine-language and the sort-pointer file
Lines 30–80	POKE in machine-language subroutine
Line 100	Defines user subroutine and clears string space
Line 110	Dimensions string array and sets U
Line 230	Finds VARPTR (variable pointer)
Line 240	In case of error
Line 250	POKEs VARPTR into memory
Line 260	POKEs number of strings into memory
Line 270	Does the sort
Line 290	Address of pointer file (always the same)

Table 1. Basic Program Lines that Are Essential

System Requirements

16K RAM
Extended Color Basic
Editor/Assembler

You can still use the program if you have more than 631 strings or if you have 16K of RAM. You have a limited amount of RAM and the computer requires not only cleared string space but also a DIM statement to set up a pointer file so it can locate stored strings.

In addition, you must PCLEAR room for your machine-language subroutine and sorting room. Table 2 helps you plan your number of strings and array dimensions, and assists you in making DIM, PCLEAR, and CLEAR changes in lines 20, 100, and 110.

Column 1 shows a 631-by-2 array DIM A$(630,1) and 20,000 bytes of string space in a 32K computer. PCLEAR 1 for sorting room.

Column 2 is for a 16K machine with a 200-by-3 array DIM A$(199,2) and 7,500 bytes for string space. Again, PCLEAR 1 for sorting room.

Column 3 illustrates what you might do if you wanted a 1,000-by-2 array DIM A$(999,1) and cleared 16500 on a 32K RAM. Notice that you would have to PCLEAR 2 for sorting room since there are over 631 strings in the array.

You are using a binary sort and can arrange strings without unnecessary comparing. It is like guessing a number from 1 to 128. You say 64. If that is too high, you say 32; too low, you say 48, and so forth.

Each guess is in the center of the remaining possible answers. You could guess the answer in, at most, seven guesses. Note that two to the seventh power is 128. To guess a number from 1 to 1,000,000 would take no more than 20 guesses. The sort is written in machine language for extra speed, but it follows the program listed in the remarks.

Notice that the strings themselves are not moved. Instead, a pointer file is used. If you have already sorted seven strings A$(0)–A$(6) and then find that A$(7) should fit in alphabetically as the fourth string, you shove V(4)–V(6) up a notch to V(5)–V(7) and make V(4) = 7.

Using the pointer file lets you ignore all the columns of your array except the one you are sorting. If you rearranged the strings themselves, you could be forced to make provisions to move the other columns as well.

The pointer file starts at &H712. When you are ready to print your list, you PEEK at &H0712 to see which string is first, &H0714 for the second string, &H0716 for the third and so forth. (Each takes 2 bytes since you must allow for over 255 strings.)

Although the statements will work fine all by themselves, the Assembly-language program is included for those who have an editor/assembler and want to know what is happening, or want to relocate to a different part of RAM. Even though the machine language is not position independent, the source code is written so you can reassemble it to another location.

"You are using a binary sort and can arrange strings without unnecessary comparing."

	COLUMN 1	COLUMN 2	COLUMN 3
System Use 0–1023 (H0000–H03FF)	1,024	1,024	1,024
Text Screen 1024–1535 (H0400–H05FF)	512	512	512
PCLEAR (Uses Multiples of 1536)			
M.L. Routine 1536–1809 (H0600–H0711)	274	274	274
Sorting Room 1810– (H0712–)	1,262	1,262	2,798
Program Storage			
(3,463 Before Deleting 30–90)	2,542	2,542	2,542
Variable Pointer File DIM A$(X,Y)			
size = 9 + 5(X + 1)(Y + 1)	6,391	3,009	10,009
String Space Reserved by Clear	20,000	7,500	15,600
Leftover RAM for Program Use and			
Other Variables	963	261	209
Total Number of Bytes	32,968	16,384	32,968

Table 2. Data Used to Plan Number of Strings and Array Dimensions

Program Listing 1. Assembly-Language Sort Routine

```
                        00120 *ML USES THE PATTERN OF THE FOLLOWING BASIC PROGRAM---
                        00130 *10 V(0)=0
                        00140 *20 FOR N=1 TO NUM-1
                        00150 *30 L=0:H=N
                        00160 *40 M=INT((H+L)/2):IFA$(N,1)<A$(V(M),1)THEN H=M:
IF H=L THEN60 ELSE 40
                        00170 *50 L=M:IF H>L+1 THEN 40
                        00180 *60 FOR P=N TO H+1 STEP -1
                        00190 *70 V(P)=V(P-1)
                        00200 *80 NEXT P
                        00210 *90 V(H)=N
                        00220 *100 NEXT N
                        00230 *------------------------
0600                    00240          ORG     $0600
0600        0000        00250 BASIC1   FDB     $0000   *POINTER FROM BASIC FOR A$(0)
0602        0000        00260 BASIC2   FDB     $0000   *FROM BASIC-# STRINGS TO SORT
0604 BE     0600        00270          LDX     BASIC1  *GET POINTER
0607 FC     0602        00280          LDD     BASIC2  *# OF STRINGS TO SORT
060A FD     0706        00290          STD     STORE1  *STORE NUM
060D CC     0712        00300          LDD     #STORE8 *ADDRESS OF V(0)
0610 FD     070C        00310          STD     STORE4  *SAVE IT
0613 CC     0000        00320          LDD     #$0000  *SET V(0)=0 (SORT FIRST STRING)
0616 ED     9F 070C     00330          STD     [STORE4]
061A CE     0001        00340          LDU     #$0001  *U=# OF STRING BEING SORTED
061D 30     05          00350 LINE1    LEAX    5,X     *ADJUST POINTER FOR A$(N)
061F FC     070C        00360          LDD     STORE4  *ADJUST ADDRESS OF FILE
0622 C3     0002        00370          ADDD    #$0002  *2 BYTES PER ADDRESS
0625 FD     070C        00380          STD     STORE4  *SAVE IT
0628 CC     0000        00390          LDD     #$0000  *MAKE L=0
062B FD     0708        00400          STD     STORE2
062E 1F     30          00410          TFR     U,D     *MAKE H=N
0630 FD     070A        00420          STD     STORE3
0633 FC     070A        00430 LINE2    LDD     STORE3  *MAKE M=H
0636 F3     0708        00440          ADDD    STORE2  *MAKE M=H+L
0639 54                 00450          LSRB            *MAKE M=(H+L)/2
063A 44                 00460          LSRA
063B 24     02          00470          BCC     LINE3
063D CB     80          00480          ADDB    #$80    *MOVE LSB OF A TO B
063F FD     070E        00490 LINE3    STD     STORE5  *STORE M
0642 CC     0712        00500          LDD     #STORE8 *FIND ADDRESS OF V(M)
0645 F3     070E        00510          ADDD    STORE5
0648 F3     070E        00520          ADDD    STORE5
064B 1F     02          00530          TFR     D,Y     *MOVE IT TO Y
064D 86     05          00540          LDA     #$05    *5 BYTES PER POINTER
064F B7     0710        00550          STA     STORE6
0652 EC     A4          00560          LDD     ,Y      *GET V(M)
0654 10BE   0600        00570          LDY     BASIC1  *POINTER FOR A$(0)
0658 1083   0000        00580          CMPD    #$0000  *?V(M)=0
065C 27     07          00590          BEQ     LINE5
065E 31     AB          00600 LINE4    LEAY    D,Y     *ADJUSTING POINTER FOR A$(V(M))
```

Listing 1 continued

Listing 1 continued

```
0660 7A   0710    00610          DEC     STORE6
0663 26   F9      00620          BNE     LINE4    *IS THIS POINTER FOR A$(V(M))?
0665 EC   02      00630 LINE5    LDD     2,X      *ADDRESS OF A$(N)
0667 FD   0701    00640          STD     HOLD1    *STORE ADDRESS OF A$(N)
066A EC   22      00650          LDD     2,Y      *ADDRESS OF A$(V(M))
066C FD   0703    00660          STD     HOLD4    *STORE ADDRESS OF A$(V(M))
066F A6   84      00670          LDA     ,X       *LENGTH OF A$(N)
0671 81   00      00680          CMPA    #0       *A$(N) BLANK?
0673 27   3C      00690          BEQ     LINE9    *IF SO A$(N)>A$(V(M))
0675 E6   A4      00700          LDB     ,Y       *LENGTH OF A$(V(M))
0677 C1   00      00710          CMPB    #0       *A$(V(M)) BLANK
0679 27   77      00720          BEQ     LINE13   *IF SO A$(N)<A$(V(M))
067B A1   A4      00730          CMPA    ,Y       *WHICH STRING IS SHORTEST?
067D 24   02      00740          BHS     LINE6
067F 1F   89      00750          TFR     A,B      *B MUST HOLD SHORTEST LENGTH
0681 F7   0705    00760 LINE6    STB     HOLD6    *STORE SHORTEST LENGTH
0684 A6   9F 0701 00770 LINE7    LDA     [HOLD1]  *A$(N) CHARACTER
0688 A1   9F 0703 00780          CMPA    [HOLD4]  *COMPARE TO A$(V(M)) CHAR
068C 26   1F      00790          BNE     LINE8    *BRANCH IF NOT THE SAME
068E FC   0701    00800          LDD     HOLD1    *ADJUST ADDRESS NEXT A$(N) CHAR
0691 C3   0001    00810          ADDD    #$0001
0694 FD   0701    00820          STD     HOLD1    *STORE IT
0697 FC   0703    00830          LDD     HOLD4    *ADJUST ADDRESS NEXT A$(V(M))CHAR
069A C3   0001    00840          ADDD    #$0001
069D FD   0703    00850          STD     HOLD4    *STORE IT
06A0 7A   0705    00860          DEC     HOLD6    LENGTH=LENGTH-1
06A3 26   DF      00870          BNE     LINE7    *GO CHECK NEXT CHARACTER
06A5 A6   84      00880          LDA     ,X       *LEN A$(N)
06A7 A1   A4      00890          CMPA    ,Y       *COMPARE TO LEN A$(V(M))
06A9 25   47      00900          BLO     LINE13   *BRANCH IF A$(N) IS SHORTER
06AB 20   04      00910          BRA     LINE9    *BRANCH
06AD 22   02      00920 LINE8    BHI     LINE9    *IF A$(N)>A$(V(M))?
06AF 20   41      00930          BRA     LINE13
06B1 FC   070E    00940 LINE9    LDD     STORE5   *MAKE L=M
06B4 FD   0708    00950          STD     STORE2
06B7 C3   0001    00960          ADDD    #$0001
06BA 10B3 070A    00970          CMPD    STORE3   *?L+1<H
06BE 1025 FF71    00980          LBLO    LINE2    *IF SO GO CHECK ANOTHER
06C2 BF   0710    00990 LINE10   STX     STORE6   *SAVE X
06C5 1F   30      01000          TFR     U,D      *FOR P=N TO H+1
06C7 B3   070A    01010          SUBD    STORE3
06CA 1F   01      01020          TFR     D,X
06CC 10BE 070C    01030          LDY     STORE4   *ADDRESS OF V(P)
06D0 1083 0000    01040          CMPD    #$0000   *ANY TO MOVE
06D4 27   0A      01050          BEQ     LINE12   *IF NOT BRANCH
```

Listing 1 continued

"In order for you to sort the strings you must know where to find them and how long each is."

In order for you to sort the strings you must know where to find them and how long each is. When the computer stores strings, it uses descriptor blocks to keep track of where it has put them. The program uses VARPTR A$(0,0) to find this descriptor block and also the fact that VARPTR (A$(N+1)) = VARPTR (A$(N)) + 5. Assume that:

A$(0,0) = "QUICK" and is stored starting at &H1E24
A$(0,1) = "SAMPLE" and is stored starting at &H1E31
A$(0,2) = "EXAMPLE" and is stored starting at &H1E4A

Listing 1 continued

```
06D6 31   3E       01060 LINE11  LEAY    -2,Y    *ADDRESS OF V(P-1)
06D8 EC   A4       01070         LDD     ,Y      *GET V(P-1)
06DA ED   22       01080         STD     2,Y     *MAKE V(P)=V(P-1)
06DC 30   1F       01090         LEAX    -1,X    *NEXT P
06DE 26   F6       01100         BNE     LINE11  *IF DONE BRANCH
06E0 BE   0710     01110 LINE12  LDX     STORE6  *GET X BACK FROM STORAGE
06E3 1F   30       01120         TFR     U,D     *GET N
06E5 ED   A4       01130         STD     ,Y      *MAKE V(H)=N
06E7 33   41       01140         LEAU    1,U     *INCREASE N
06E9 11B3 0706     01150         CMPU    STORE1  *DONE YET
06ED 1026 FF2C     01160         LBNE    LINE1   *IF NOT BRANCH
06F1 39            01170         RTS             *GO BACK TO BASIC PROGRAM
06F2 FC   070E     01180 LINE13  LDD     STORE5  *GET M
06F5 FD   070A     01190         STD     STORE3  *MAKE H=M
06F8 10B3 0708     01200         CMPD    STORE2  *IS H=L
06FC 27   C4       01210         BEQ     LINE10  *IF SO GO ADJUST MIDDLE
06FE 16   FF32     01220         LBRA    LINE2   *IF NOT BRANCH
0701      0000     01230 HOLD1   FDB     $0000   *ADDRESS OF A$(N)
0703      0000     01240 HOLD4   FDB     $0000   *ADDRESS OF A$(V(M))
0705      00       01250 HOLD6   FCB     $00     *LENGTH OF STRING
0706      0000     01260 STORE1  FDB     $0000   *NUMBER OF STRINGS TO SORT
0708      0000     01270 STORE2  FDB     $0000   *VARIABLE L-LOWER LIMIT
070A      0000     01280 STORE3  FDB     $0000   *VARIABLE U-UPPER LIMIT
070C      0000     01290 STORE4  FDB     $0000   *ADDRESS OF V(N)
070E      0000     01300 STORE5  FDB     $0000   *VARIABLE M-MIDDLE
0710      0000     01310 STORE6  FDB     $0000   *STORAGE FOR X
0712      0000     01320 STORE8  FDB     $0000   *FIRST BYTE OF V(0)
          0000     01330         END
00000 TOTAL ERRORS

BASIC1  0600          LINE12  06E0          LINE8   06AD
BASIC2  0602          LINE13  06F2          LINE9   06B1
HOLD1   0701          LINE2   0633          STORE1  0706
HOLD4   0703          LINE3   063F          STORE2  0708
HOLD6   0705          LINE4   065E          STORE3  070A
LINE1   061D          LINE5   0665          STORE4  070C
LINE10  06C2          LINE6   0681          STORE5  070E
LINE11  06D6          LINE7   0684          STORE6  0710
                                            STORE8  0712
                                                        END
```

Then if VARPTR(A$(0,0)) = &H2E00 you would find the following if we examined RAM:

2E00	5	Length of A$(0,0)
2E01	-	Reserved for system use
2E02	1E	2-byte address of where A$(0,0)
2E03	24	is stored
2E04	-	Reserved for system use
2E05	6	Length of A$(0,1)
2E06	-	Reserved for system use
2E07	1E	2-byte address of where A$(0,1)
2E08	31	is stored
2E09	-	Reserved for system use
2E0A	7	Length of A$(0,2)
2E0B	-	Reserved for system use
2E0C	1E	2-byte address of where A$(0,2)
2E0D	4A	is stored
2E0E	-	Reserved for system use

This permits you to identify exactly where each string is stored and lets you compare them one letter at a time so you can sort them. ■

Address correspondence to Anna M. Reeves, Rte. 2, Box 10 R9, Espanola, WA 99022.

```
10 'ANN REEVES-USING A ML BINARY
 SORT TO ARRANGE A HAT COLLECTIO
N
20 PCLEAR1 ''1=UP TO 631 STRINGS
,2=UP TO 1399, 3=2167, 4=2935
30 CLS:PRINT"WAIT WHILE THE MACH
INE LANGUAGE SUBROUTINE IS POKED
 INTO PLACE"
40 FOR Y=&H6000 TO &H7000:READ A$
:POKEY,VAL("&H"+A$):NEXTY
50 DATA 0,0,0,0,BE,06,00,FC,06,0
2,FD,07,06,CC,07,12,FD,07,0C,CC,
00,00,ED,9F,07,0C,CE,00,01,30,05
,FC,07,0C,C3,00,02,FD,07,0C,CC,0
0,00,FD,07,08,1F,30,FD,07,0A,FC,
07,0A,F3,07,08,54,44,24,02,CB,80
,FD,07,0E,CC,07,12,F3,07,0E,F3,0
7,0E,1F,02,86,05,B7
60 DATA 07,10,EC,A4,10,BE,06,00,
10,83,00,00,27,07,31,AB,7A,07,10
,26,F9,EC,02,FD,07,01,EC,22,FD,0
7,03,A6,84,81,00,27,3C,E6,A4,C1,
00,27,77,A1,A4,24,02,1F,89,F7,07
,05,A6,9F,07,01,A1,9F,07,03,26,1
F,FC,07,01,C3,00,01,FD,07,01,FC
65 DATA 07,03,C3,00,01,FD,07,03,
7A
70 DATA 07,05,26,DF,A6,84,A1,A4,
25,47,20,04,22,02,20,41,FC,07,0E
,FD,07,08,C3,00,01,10,B3,07,0A,1
0,25,FF,71,BF,07,10,1F,30,B3,07,
0A,1F,01,10,BE,07,0C,10,83,00,00
,27,0A,31,3E,EC,A4,ED,22,30,1F,2
6,F6,BE,07,10,1F,30,ED,A4,33,41
75 DATA 11,B3,07,06,10,26,FF,2C,
39
80 DATA FC,07,0E,FD,07,0A,10,B3,
07,08,27,C4,16,FF,32
90 DEL 30-90
100 DEFUSR1=&H0604:CLEAR 20000''
''CLEAR SUFFICIENT STRING SPACE-
--DEPENDS ON PROGRAM LENGTH, RAM
, AND NUMBER OF STRINGS
110 DIM A$(630,1):U=630'''U MUST
 BE THE SAME AS THE FIRST NUMBER
 IN THE DIM STATEMENT
120 CLS:PRINT"<1> BRAND NEW HAT
COLLECTION","<2> ADD HATS TO COL
LECTION","<3> CORRECT HAT INFORM
ATION","<4> DELETE HAT FROM COL
LECTION","<5> ENTER COLLECTION F
ROM TAPE","<6> SAVE COLLECTION O
N TAPE"
130 PRINT"<7> LIST COLLECTION-HA
T ORDER","<8> LIST COLLECTION-SO
URCE ORDER
140 INPUT A:ON A GOTO 160,220,53
0,480,420,380,230,610
150 GOTO 140
160 CLS:INPUT "SURE????-OLD LIST
 WILL ERASE       <YES> - <NO>"
;A$:IF A$<>"YES"THEN 120
170 NUM=0:FOR X=0 TO U-1
180 INPUT "HAT NAME";Y$:A$(X,0)=
Y$
190 INPUT "SOURCE";X$:A$(X,1)=X$
:NUM=NUM+1
200 INPUT "MORE <Y> - <N>";A$:IF
 A$<>"Y"THEN 120
210 NEXT X:GOTO 120
220 IF NUM=0 THEN 120 ELSE FOR X
=NUM TO U-1:GOTO 180
230 S=VARPTR(A$(0,0))'SORTING BY
 FIRST COLUMN---CAUTION!!!DO NOT
 INTRODUCE ANY NEW VARIABLES OR
STRINGS IN BETWEEN THE TIME YOU
FIND S=VARPTR AND WHEN YOU USE
X=USR1(0) FUNCTION TO PERFORM
THE SORT!!!
240 IF NUM<2 THEN INPUT"NEED MOR
E STRINGS TO ARRANGE    <ENTER>"
;C$:GOTO 120
250 PRINT"SORTING":POKE &H0600,I
NT(S/256):POKE &H0601,S-INT(S/25
6)*256
260 POKE&H0602,INT(NUM/256):POKE
&H0603,NUM-INT(NUM/256)*256
270 X=USR1(0)
280 INPUT"HOLD THE <A> KEY DOWN
TO ABORT--ON PRINTER <P> OR SCRE
EN <S>";B$
290 X=&H0712''ADDRESS OF POINTER
300 FOR N=0 TO NUM-1
310 P=256*PEEK(X)+PEEK(X+1)
320 IF B$="P" THEN 370
330 PRINT A$(P,0)" - "A$(P,1)
340 X=X+2:C$=INKEY$:IF C$="A" TH
EN 120
350 NEXT N
360 INPUT "<ENTER> FOR MENU";B$:
GOTO 120
370 PRINT#-2,N+1 TAB(5)A$(P,0)TA
B(35)" - "A$(P,1):GOTO340
380 INPUT "TAPE READY <YES>-<NO>
";C$:IF C$<>"YES" THEN 120 ELSE
OPEN"O",#-1,"DATA":FOR X=0 TO NU
M-1
390 PRINT#-1,A$(X,0),A$(X,1)
400 NEXT X
410 CLOSE:GOTO 120
420 INPUT "TAPE READY <YES>-<NO>
";C$:IF C$<>"YES"THEN120 ELSE NU
M=0
430 OPEN"I",#-1,"DATA"
440 FOR X=0 TO U-1
450 IF EOF(-1)THEN470 ELSEINPUT#
-1,A$(X,0),A$(X,1): NUM=NUM+1
460 NEXTX
470 CLOSE:GOTO120
480 CLS:INPUT"NAME OF HAT TO DEL
ETE";B$
490 FOR X=0 TO NUM-1:IF A$(X,0)=
B$ THEN GOTO 510
500 NEXT X:INPUT"NOT FOUND <ENTE
R> FOR MENU";C$:GOTO 120
510 FOR Y=X TO NUM-2:A$(Y,0)=A$(
Y+1,0):A$(Y,1)=A$(Y+1,1):NEXTY
520 NUM=NUM-1:GOTO 120
530 CLS:INPUT "NAME OF HAT YOU W
ISH TO CORRECT";B$
540 FOR X=0 TO NUM-1:IF A$(X,0)=
B$THEN 560
550 NEXT X:PRINT "NOT FOUND":INP
UT"<ENTER> FOR MENU";B$:GOTO 120

560 PRINT "HAT--"A$(X,0):PRINT"S
OURCE--"A$(X,1)
570 PRINT"<C> TO CHANGE INFO","<
S> SEARCH REST OF FILE","<M> GO
BACK TO MENU":INPUT C$
580 IF C$="M" THEN GOTO 120
590 IF C$="S" THEN GOTO 500
600 INPUT "CORRECT HAT NAME";C$:
A$(X,0)=C$:INPUT"CORRECT SOURCE"
;C$:A$(X,1)=C$:GOTO 120
610 S=VARPTR(A$(0,1)):GOTO 240''
SORTING BY SECOND COLUMN
```

Program Listing 2. Basic Data-Management Program

END

BY BRUCE SACHETTI

TURBO-STICK

Modify your hardware and software to end joystick and screen resolution incompatibility problems.

Incompatibility between joystick and screen resolution is one of the few drawbacks of the Color Computer. But you can increase joystick resolution until it is similar to the screen resolution. Since it must be the same for both the horizontal and vertical directions, and it must be a base-two number, the most logical resolution is 256 by 256. To achieve this you must perform hardware and software modifications.

Hardware

The modification requires very little hardware and what you use attaches to the board without soldering or cutting any traces on the board. You need a wire-wrap tool, 30-gauge wire, solder, soldering gun, Phillips-head screwdriver, 640K resistor, 1.28M resistor, and a 74LS02 integrated circuit. Though resistors are difficult to find in these specific values, close approximations of the appropriate values also work.

I use a 600K resistor and a 1M resistor with very good results. If you do not have a 74LS02, any chip with two inverters or two buffers will work. If you use buffers, you need a software change (covered in a later section).

First, open the computer by removing the screws on the bottom of the case. (This act will void the official Radio Shack warranty.) Next, solder the two resistors on the integrated circuit. Solder the 640K resistor to an output of the chip, pin 1 on a 74LS02, and the 1.28M resistor to an output of the chip, pin 13 on a 74LS02.

The next step is to place the integrated circuit inside the computer. To do this you can place a piece of tape beneath the keyboard and glue the integrated circuit to the tape, upside-down. Tie the two loose ends of the resistors together with one end of a wire and solder. Now, wire-wrap the other end of

System Requirements
16K RAM
Extended Color Basic

See List of Advertisers on page 99

this wire to test point 7 on the main board. Next, run the power leads for the integrated circuit. Wire-wrap a wire from test point 3 to the ground on the chip (on a 74LS02 that is pin 7). Also, wire-wrap a wire from test point 12 to the 5V supply on the chip (on a 74LS02 that is pin 14).

Only two wires remain to be connected from the keyboard connector to inputs of the chip. The keyboard connector is numbered from 1 on the left to 16 on the right as you face the computer. You might have to bend the metal shield back away from the connector for more room to wire-wrap.

"This modification makes your joystick and high-resolution graphics compatible."

Connect a wire from pin 9 on the keyboard connector to the input of the buffer where the 1.28M resistor is connected (pins 11 and 12 on the 74LS02). Next, connect a wire from pin 10 of the keyboard connector to the input of the buffer where the 640K resistor is connected (pins 2 and 3 on the 74LS02). This completes the modification.

Now, reassemble the computer and use the program in the software section to test the circuit. Hardware hackers might want to place the resistors on the unused gates of the 74LS02 (on the main board of the computer). This way you don't need to run power leads to a spare chip. The only problem is that on different computers different gates are used in this chip, so it is important to know your computer.

Software

The software for this modification is written in two different forms. The Assembly-language version of the program, for the advanced user, is in Listing 1. For easier understanding the program includes comments and can be assembled into any memory and executed to perform exactly the same function as the second form of the program.

Listing 2 is a small Basic routine that can be incorporated in any Basic program or used by itself.

To use this small program, type it in just after turning on your computer. It has many features to reduce the possibility of user problems. It will function on any machine regardless of the Basic

Program Listing 1. Assembly-Language Version

```
0200            00180 ORG    $200        PROGRAM  IS FULLY RELOCATABLE,  MAY  BE
CHANGED, $200, $600 OR $E00 RECOMMENDED
        006D    00190 LENGTH  EQU  END-CMD     CALCULATE LINGTH OF JOY PROGRAM
0200 DC 21      00200         LDD  <$21       GET TOP OF STACK
0202 83 006D    00210         SUBD #LENGTH    SUBTRACT LENGTRH OF JOY PROGRAM
0205 25 0B      00220         BLO  OMERR      BRANCH IF CARRY TO OM ERROR
0207 1F 03      00230         TFR  D,U        SAVE CORRECT NUMBER
0209 83 003A    00240         SUBD #$3A       ALLOW FOR STACK AREA
020C 25 04      00250         BLO  OMERR      BRANCH IF NOT ENOUGH MEMORY
020E 93 1B      00260         SUBD <$1B       SUBTRACT LOWER MEMORY USED
0210 24 03      00270         BHS  NOOM       BRANCH IF ENOUGH MEMORY
0212 7E AC44    00280 OMERR   JMP  >$AC44     INDICATE  OM  ERROR AND RETURN TO
BASIC
0215 DF 21      00290 NOOM    STU  <$21       RESTORE NEW TOP OF STACK
0217 9E 27      00300         LDX  <$27       GET TOP OF STRING AREA
0219 30 88 93   00310         LEAX <-LENGTH,X SUBTRACT LENGTH OF JOY PROGRAM
021C 9F 27      00320         STX  <$27       RESTORE NEW TOP OF STRING AREA
021E 30 01      00330         LEAX 1,X        X NOW POINTS TO NEW FREE AREA
0220 33 8C 34   00340         LEAU <CMD,PCR   GET START OF JOY PROGRAM
0223 34 10      00350         PSHS X          SAVE START OF JOY
0225 A6 C0      00360 MOVE    LDA  ,U+        GET JOY PROGRAM
0227 A7 80      00370         STA  ,X+        STORE PROGRAM IN HIGH MEMORY
0229 81 43      00380         CMPA #$43       CHECK IF PROGRAM MOVED
022B 26 F8      00390         BNE  MOVE       BRANCH IF MORE TO BE MOVED
022D 35 10      00400         PULS X          RESTORE START OF JOY
022F CE  0143   00410           LDU   #$143       GET ADDRESS  OF  DISK  BASIC
FUNCTION EXPANSION TABLE
0232 CC  444B   00420           LDD  #$444B     'DK'  DATA TO  SEE IF DISK BASIC
PRESENT
0235 10B3 C000  00430         CMPD >$C000     CHECK FOR DISK ROM
0239 27  0D     00440         BEQ  DSKEXT     BRANCH IF DISK ROM PRESENT
023B 33  56     00450           LEAU -10,U      GET  ADDRESS OF EXTENDED  BASIC
FUNCTION EXPANSION TABLE
023D CC  4558   00460           LDD   #$4558      'EX' DATA TO  SEE  IF  EXTENDED
BASIC PRESENT
0240 10B3 8000  00470         CMPD >$8000     CHECK FOR EXTENDED BASIC ROM
0244 27  02     00480           BEQ   DSKEXT     BRANCH  IF  EXTENDED BASIC  ROM
PRESENT
0246  33  56     00490          LEAU -10,U      GET  ADDRESS OF STANDARD FUNCTION
EXPANSION TABLE
0248 86  01     00500 DSKEXT  LDA  #$1         NUMBER OF FUNCTIONS TO ADD
024A A7  C0     00510         STA  ,U+        UPDATE FUNCTION TABLE
024C AF  C1     00520         STX  ,U++       SET UP FUNCTION NAME TABLE
024E 30  07     00530         LEAX 7,X        GET EXECUTE ADDRESS OF JOY
0250 AF  C1     00540         STX   ,U++      SET UP  FUNCTION  EXECUTE ADDRESS
TABLE
0252 6F  45     00550         CLR  5,U        CLOSE FUNCTION TABLE
0254 7E  AD26   00560          JMP  >$AD26     PERFORM BASIC STATUS MANIPULATION
AND RETURN
0257     4A     00570 CMD     FCC  /JO/       NAME OF NEW FUNCTION 'JOY'
         4F
0259     D9     00580         FCB  $D9        EXAMPLE OF USE 'A=JOY(0)'
025A     0000   00590  TABLE   FDB   0          TABLE  FOR  STORAGE  OF JOYSTICK
POSITIONS
025C     0000   00600         FDB  0
025E BD  B262   00610 PROG    JSR  >$B262     EVALUATE PARENTHESES
0261 BD  B70E   00620         JSR  >$B70E     EVALUATE JOY ARGUMENT
0264 C1  03     00630         CMPB #$3        CHECK IF VALID JOYSTICK NUMBER
0266 23  03     00640         BLS  NOERR      BRANCH IF VALID
0268 7E  B44A   00650         JMP  >$B44A     DIPLAY FC ERROR
026B 5D         00660 NOERR   TSTB            CHECK IF SAMPLING MUST OCCUR
026C 26  49     00670         BNE  NOSAM      BRANCH IF JOYSTICK NUMBER NOT 0
026E BD  A974   00680         JSR  >$A974     TURN OFF AUDIO
0271 30  8C EA  00690         LEAX TABLE+4,PCR GET END OF VALUE TABLE
0274 C6  03     00700         LDB  #$3        NUMBER OF ELEMENTS IN TABLE
0276 86  0A     00710 NXTJOY  LDA  #10        MAXIMUM SAMPLES FOR EACH
0278 ED  E3     00720         STD  ,--S       SAVE VALUES FOR LATER
027A BD  A9A2   00730         JSR  >$A9A2     SELECT THE INPUTS ON MULTIPLEXE
027D CC  4080   00740 NXTTRY  LDD  #$4080     COMPARISON VALUES
0280 A7  E2     00750 NXTAPX  STA  ,-S        SAVE A FOR LATER CALCULATIONS
0282 34  04     00760         PSHS B          SAVE B TEMPORARILY
0284 C4  FC     00770         ANDB #$FC       MASK OFF 6 BITS
0286 CA  02     00780         ORB  #$2        SET UP OTHER BITS
0288 F7  FF20   00790         STB  >$FF20     OUTPUT VALUE TO D/A CONVERTER
028B 35  04     00800         PULS B          RECOVER GOOD DATA
028D 53         00810         COMB            INVERT BITS (SEE TEXT)
028E F7  FF02   00820         STB  >$FF02     OUTPUT OTHER 2 BITS
0291 53         00830         COMB            RESTORE DATA (SEE TEXT)
0292 7D  FF00   00840         TST  >$FF00     CHECK COMPARISON BIT
0295 2B  08     00850         BMI  ADD        INCREASE APPROXIMATE VALUE
0297 6D  E4     00860         TST  ,S         AT END OF BYTE
0299 26  01     00870         BNE  NOCECB     BRANCH OVER DECREMENT
029B 5A         00880         DECB            FOR LAST SAMPLE
029C E0  E4     00890 NODECB  SUBB ,S         CREATE NEW APPROXIMATION
029E     8C     00900         FCB  $8C        SKIP NEXT INSTRUCTION
029F EB  E4     00910 ADD     ADDB ,S         CREATE NEW APPROXIMATION
02A1 A6  E0     00920         LDA  ,S+        DONE APPROXIMATING?
02A3 27  03     00930         BEQ  NOAPX      BRANCH IF DONE
02A5 44         00940         LSRA            NEW OFFSET VALUE
02A6 20  D8     00950         BRA  NXTAPX     APPROXIMATE AGAIN
02A8 E1  1F     00960 NOAPX   CMPB -1,X       CHECK FOR CONSISTENT SAMPLE
02AA 27  04     00970         BEQ  GOODD      BRANCH IF GOOD DATA
02AC   6A     E4     00980           DEC     ,S           DECREMENT   NUMBER
APPROXIMATIONS
02AE 26  CD     00990         BNE  NXTTRY     BRANCH IF MORE ALLOWABLE
02B0 E7  82     01000 GOODD   STB  ,-X        SAVE GOOD DATA IN TABLE
02B2 EC  E1     01010         LDD  ,S++       RECOVER SAMPLING STATUS
02B4 5A         01020         DECB            DECREMENT JOYSTICK NUMBER
02B5 2A  BF     01030         BPL  NXTJOY     CONTINUE IF MORE JOYSTICKS
02B7 30  8C A0  01040 NOSAM   LEAX TABLE,PCR  GET ADDRESS OF VALUES
02BA D6  53     01050         LDB  <$53       GET JOYSTICK NUMBER
02BC E6  85     01060         LDB  B,X        GET APPROPRIATE DATA
02BE BD  B4F3   01070         JSR  >$B4F3     GET NEXT COMMAND
02C1 7E  B143   01080         JMP  >$B143     PROCESS MORE PROGRAM
```

Listing continued

See List of Advertisers on page 99

Listing continued

```
02C4      42    01090 END     FCC  /BY D. NELSON/
          59
          20
          44
          2E
          20
          4E
          45
          4C
          53
          4F
          4E
          0200  01100         END  $200       MAY BE CHANGED, SEE ORIGIN
00000 TOTAL ERRORS
```

END

```
10 FORA=512TO719
20 READB
30 POKEA,B
40 NEXTA
50 EXEC512
60 NEW
70 DATA220,33,131,0,109,37,11,31,
3,131,0,58,37,4,147,27,36,3,126,
172,68,223,33,158,39,48,136,147,
159,39,48,1,51,140,52,52,16,166,
192,167,128,129,67,38,248,53,16,
206,1,67,204,68,75
80 DATA16,179,192,0,39,13,51,86,
204,69,88,16,179,128,0,39,2,51,8
6,134,1,167,192,175,193,48,7,175
,193,111,69,126,173,38,74,79,217
,0,0,0,0,189,178,98,189,183,14,1
93,3,35,3,126,180,74
90 DATA93,38,73,189,169,116,48,1
40,234,198,3,134,10,237,227,189,
169,162,204,64,128,167,226,52,4,
196,252,202,2,247,255,32,53,4,83
,247,255,2,83,125,255,0,43,8,109
,228,38,1,90,224,228
100 DATA140,235,228,166,224,39,3
,68,32,216,225,31,39,4,106,228,3
8,205,231,130,236,225,90,42,191,
48,140,160,214,83,230,133,189,18
0,243,126,177,67,66,89,32,68,46,
32,78,69,76,83,79,78
```

Program Listing 2. Basic Version

or the size of RAM. It takes only the program that is absolutely necessary, and places it in upper memory. It then deletes itself to conserve memory, and requires no CLEAR statement to set the upper memory limits, since the program performs this function itself. The resulting program is placed between the string area and the cleared area, with the string area being moved down.

This program is only 109 bytes long, so place other information in the cleared area before running the program. This reduces the possibility of problems.

The procedure for finding the joystick values is similar to the joystick routine in Basic. In Basic the JOYSTK(X) command, where X is zero to three, works, but the new command JOY(X), where X is zero to three, has been added, and it returns a range from 0 to 255. For example, enter Program Listing 3 after running the joystick program.

If you use a buffer instead of an inverter, change the two COMB instructions in Listing 1 to NOP instructions. The two COMB instructions have the note "see text" in the comment column for identification. In Listing 2 the DATA statement in line 90 has the following set of bytes: 83, 247, 255, 2, 83. These must be changed to 18, 247, 255, 2, 18, to allow for the use of buffers.

This modification makes your joystick and high-resolution graphics compatible. It is a feature that is not found on other computers and something to be proud of. ■

Address correspondence to Bruce Sachetti Jr., 17702 East Kansas Pl., Aurora, CO 80017.

```
10 CLS
20 PRINT@0,JOY(0),JOY(1)
30 PRINTJOY(2),JOY(3)
40 IFPEEK(32768)<>69THEN20
50 PMODE4,1
60 PCLS
70 SCREEN1,1
80 H0=JOY(0)
90 V0=JOY(1)
100 H1=JOY(2)
110 V1=JOY(3)
120 H2=JOY(0)
130 V2=JOY(1)
140 H3=JOY(2)
150 V3=JOY(3)
160 LINE(H0,V0)-(H2,V2),PSET
170 LINE(H1,V1)-(H3,V3),PSET
180 H0=H2
190 V0=V2
200 H1=H3
210 V1=V3
220 IF(PEEK(65280)AND3)<>3THENPC
LS
230 GOTO120
```

Program Listing 3. Example Joystick Program

1983 unit sales	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total	Average	Best	Worst
Bach	136	139	119	161	130	104	84	121	95	115	75	161	1440		161	75
Chalone	120	170	152	170	182	102	89	157	162	129	64	158			182	64
Dolan	188	157	103	112	161	122	99	145	145	103					188	97
Feagan	105	94	127	115	157	97	61	132	113						174	61
Graham	135	135	183	116	151	104	86	149							183	63
Harpel	134	102	190	161	180	85										
Jordan	105	109	188	171	120											
Latour	112	128	124	129												
Lucido	158	110														
Phelps	167															
Prats																75
Schaeferle															193	78
Taylor														145	190	88
Torres					131								1620	135	177	105
Turner				127	131						75	178	1635	136	190	75
Wehlen			145	142	154			137	125	106	60	151	1495	125	154	60
	2312	2166	2387	2321	2401	1699	1439	2276	2242	2011	1318	2631	25203	2100		

DYNACALC®
FOR CoCo DOS
NO OTHER OPERATING SYSTEM NEEDED!

THE BEST OF BOTH WORLDS!

available from

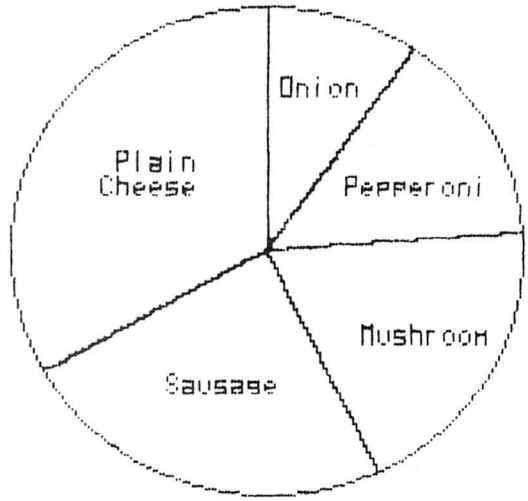
MasterCard VISA

COMPUTER SYSTEMS CENTER
13461 Olive Blvd.
Chesterfield, MO 63017 USA
(314) 576-5020

CSC

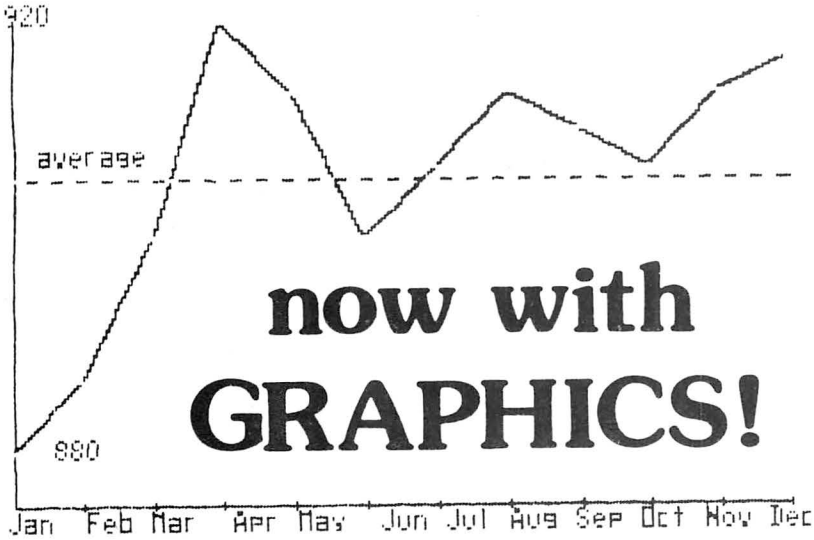
506

or your local DYNACALC dealer

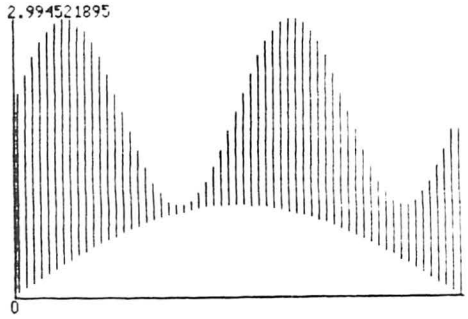


Price $150 postpaid in US & Canada.
Outside North America add $10 postage.

DYNACALC Reg. U. S. Pat. Off.

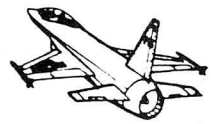


now with GRAPHICS!



CANADA
MICRO R.G.S. INC.
751, CARRE VICTORIA, SUITE 403
MONTREAL, QUEBEC, CANADA, H2Y 2J3
Regular Tel. (514) 287-1563
Canadian Toll Free 800-361-5155

See List of Advertisers on page 99



F-16 INSTRUMENT FLIGHT SIMULATOR

An exciting simulation of instrument flight with full panel and "heads up" display. Specify 16 or 32/64 K $21.95 + $1.00 S&H

TSPOOL

A software spooler for Telewriter-64. Optional descenders for DMP-100 / Gorilla Banana printers. For 32/64K specify printer $24.95 + 1.00 S&H

DESCENDERS

Graphic lowercase descenders for LPVII /DMP-100 printers. For use with Basic (16/32/64K) or Telewriter-64 (32/64K) only. $17.95 + $1.00 S&H

TELEWRITER-64

The BEST word processor you can buy. Save 15% on our TSPOOL or Descenders if you order in combo with Telewriter-64. $49.95 Tape $59.95 disk

COD and CREDIT CARD orders gladly accepted
9am to 5pm EST (813) 321-2840

MasterCard VISA

KRT Software Inc
P O Box 41395
St Petersburg, Fl 33743

↙395

DISCOUNT PRICES COMPUTERS & EQUIPMENT

COMPUTERS

SANYO

FRANKLIN
ACE 1000 COMPUTER
ACE 1200 COMPUTER

IBM COMPATIBLES YOUR CHOICE

EAGLE COMPUTER

COLUMBIA DATA PRODUCTS, INC.

TRS-80 COMPUTERS

EQUIPMENT

Hayes · Novation

QUADRAM · AMDEK

PGS Princeton Graphic Systems · PLANTRONICS

Verbatim · AST RESEARCH INC.

PRINTERS

EPSON · C. Itoh · star

PRINTER CARDS AND CABLES

OKIDATA · TOSHIBA · MANNESMANN TALLY

Orange Micro Inc.

Panasonic. · daisywriter · Transtar · cardco, inc.

JUKI · SCM SMITH-CORONA · BOTEK

Since 1978

FREE
Price List & Information Kit

Call For Your Discount Prices

Micro Management Systems, Inc.
2803 Thomasville Road East
Cairo, Georgia 31728
(912) 377-7120

↙96

1-800-841-0860

```
**********************************************************
*                                                        *
*          * 96K EXPANDERS *                             *
*                                                        *
* Use all of your memory in 64K Computers.  The 96KX     *
* is in permanent ROM & occupies the upper 8K.  Com-     *
* patible with disk drives & accessories.  Module        *
* mounts inside computer & cartridge is for tape or      *
* multipack selectors.  96KX-M $59.95, 96KX-C $49.95     *
*                                                        *
*        * VIDEO REVERSER *                              *
* Provides (1) Reversed, (2) Reversed all capitals,      *
* & (3) Normal.  Solderless installation.  $19.95        *
*                                                        *
*     * DYNAMIC COLOR NEWS *                             *
*                                                        *
* A monthly engineering newsletter that explains how     *
* the Color Computers work in nontechnical terms.        *
* Includes Basic & Machine Language Programming, ex-     *
* pansion techniques, questions & answers, etc. Re-      *
* ceive discounts on our products. $15/yr, Sample $1     *
*                                                        *
*          MEMORY EXPANDERS                              *
*      No trace cutting, solderless, & reversible.       *
* ME-3 . . . D, E, & 285 from 16K to 32K       $36.95    *
* ME-4 . . . D & E computers to 64K            $89.95    *
* ME-4F  . . F or 285 and COCO 2 to 64K        $79.95    *
* ME-128-64 . 64K computers to 128K           $199.00    *
* We will loan you a computer while we upgrade yours.    *
*                                                        *
* DISASM - Decimal Disassembler & Assembler    $19.95    *
* DYTERM - 300 to 2400 baud Terminal Program   $14.95    *
* MPM - Stack 5 programs in 32K computers      $14.95    *
*                                                        *
* New 64K CoCo 2 with 96KX & video reverser   $325.00    *
* Used 128K TDP-100 with 96KX & video rev.    $395.00    *
*                                                        *
* Free catalog. Checks, VISA & MC cards. Add $2 ship.    *
*  24 hr. phone. Call at nights & on weekends & save     *
*                                                        *
*  DYNAMIC ELECTRONICS INC.                              *
*  Box 896   (205) 773-2758)                             *
*     Hartselle, AL  35640                               *
*                                                ↙72     *
**********************************************************
```

PRO-COLOR-FILE *ENHANCED*

↙35

PRO-COLOR-FILE has become one of the most respected database programs ever developed for the Color Computer. Whether it's for home or business, **PRO-COLOR-FILE** lets you design your own sophisticated database tailored to your needs.

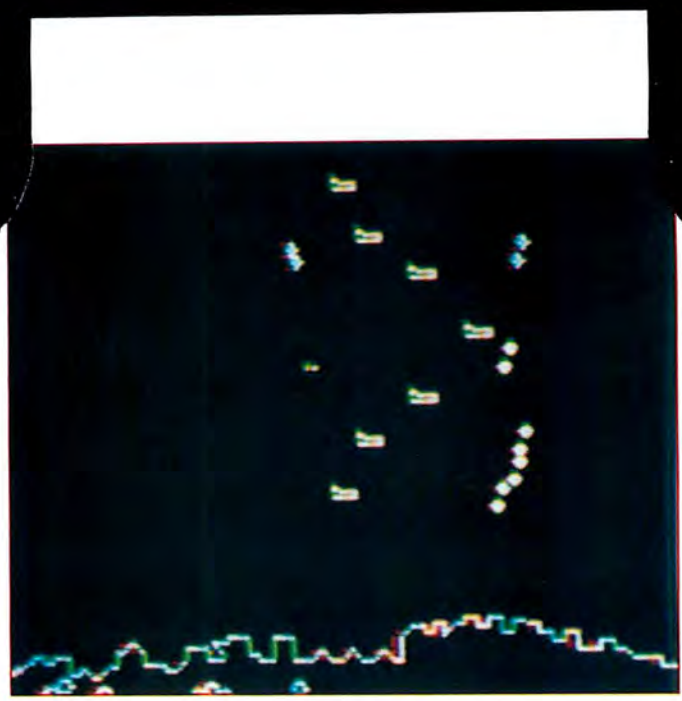
60 DATA FIELDS available for each record to store information
1020 BYTES for each record can be used if needed
1 - 4 DISK DRIVES can be used to maximize storage capacity
4 COLOR DATA ENTRY screens can be custom designed
28 MATH EQUATIONS can be setup to perform calculations
POST ACCOUNTS routine performs calculations on an entire file
DUPLICATE RECORDS or FIELDS from previous entries
SORT ENTIRE FILE on 3 fields at one time
SORT ANY SIZE FILE whether it's 200 or 2000 records
SELECT SUB-SETS of file for sorting or reporting
SCAN FILES alphabetical by any field
SUMMARIZE FILES to find totals, averages, low and hi values
8 REPORT FORMATS for obtaining hard or soft copy reports
6 LABEL FORMATS for 1 to 10 across labels & 1 to 30 lines/label
PASSWORD PROTECTION for limited access to data and reports

PRO-COLOR-FILE was so well received that over 70% of the owners of the original version ordered the **PRO-COLOR-FILE *ENHANCED*** upgrade in the first 2 weeks of its introduction. Find out for yourself why people in almost every state, Canada, South America, and even the Far East and Europe have turned to **PRO-COLOR-FILE *ENHANCED*** for their information management needs.

PRO-COLOR-FILE *ENHANCED* **$79.95**

See your local dealer or send check or money order to:
DERRINGER SOFTWARE, INC., P.O. Box 5300, Florence, South Carolina 29502. Visa/MC customers call (803)665-5676
Add $3.50 S&H - Available on AMDISK (Add $5.00).
S.C. residents add required sales tax.

✓ See List of Advertisers on page 99



Arcade Action

Without Machine Language

Programming Technique

by Gary Wick

One of the many advantages of having Extended Color Basic is the ability to create screen images using only a few commands. You can also move images across the screen at a reasonable rate to simulate arcade-game action. All of this can be done without using difficult machine language.

Save London is an arcade game that demonstrates Extended Basic's capabilities in creating graphics in motion. But before you type in Save London, look at a few simple examples of Extended Basic in action.

Type in Program Listing 1, run it, and you should observe a space shuttle landing, at night, on a flat surface. Listing 1 uses high-resolution graphics and provides movement of the space shuttle using GET and PUT commands to get an image that you drew and put it someplace else on the screen. This article demonstrates and experiments with Extended Basic commands, so I won't go into detail concerning GET and PUT.

Listing 1 has a major flaw: too much flickering, caused by putting the space shuttle at a certain location on the screen, putting a black square over the shuttle to erase it, and then putting the shuttle in a new location. This process continues until the shuttle lands. Eliminate the flickering by using Program Listing 2.

Type in Listing 2. The array that holds the pixel image of the shuttle, array A, is increased when you first DIM A (5,5) in the beginning of Listing 2.

```
5 'listing 1
10 ' DRAW SPACE SHUTTLE
20 DIM A(1),B(1)
30 PMODE 4,1: PCLS: SCREEN 1,1
40 DRAW"BM12,10;D2R8"
50 DRAW"BM12,10;F1R7"
60 LINE(20,11)-(22,12),PSET
70 DRAW"BM0,140;R252"
80 GET(0,0)-(4,4),B,G
90 'LAND SPACE SHUTTLE
100 IH=0:IV=0
110 GET(8,10)-(22,13),A,G
120 PUT(8,10)-(22,13),B,PSET
130 PLAY"L100;A;C;E;O1"
140 IH=IH+3
150 IV=IV+3
160 PUT(8+IH,10+IV)-(22+IH,13+IV
),A,PSET
170 PUT(8+IH,10+IV)-(22+IH,13+IV
),B,PSET
180 IFIH=126 THEN GOTO 200
190 GOTO130
200 PUT(8+IH,10+IV)-(22+IH,13+IV
),A,PSET:FORX=1TO1000:NEXTX
210 GOTO210
```

Program Listing 1. Draw Space Shuttle, with Flicker

Since you have more room in array A, you can include a large black border. As you PUT array A, the image of the space shuttle, across the screen, the black border erases the previous image as the space shuttle moves across the screen to its landing. The flickering is gone and you have smoother action.

Change PSET to PRESET at the end of the PUT command, line 40, and observe array A's border that surrounds the space shuttle. In Listing 2 you have also added another nice touch: sound.

You can make sound effects with the PLAY command, and by changing the tempo, length of the note, octave, note value, and volume, a large number of sound effects are at your fingertips without using machine language. Program Listing 3 gives us a sample of sound effects using a FOR...NEXT loop and the PLAY command. You needn't be a music major to master this command. Play around with it, and soon you'll be making your own sound combinations.

```
0 'listing 2
10 ' DRAW SPACE SHUTTLE
20 DIM A(5,5)
30 PMODE 4,1: PCLS: SCREEN 1,1
40 DRAW"BM12,10;D2R8"
50 DRAW"BM12,10;F1R7"
60 LINE(20,11)-(22,12),PSET
70 DRAW"BM0,140;R252"
80 'LAND SPACE SHUTTLE
90 IH=0:IV=0
100 GET(5,5)-(22,13),A,G
110 PLAY"L100;A;C;E;O1"
120 IH=IH+3
130 IV=IV+3
140 PUT(5+IH,5+IV)-(22+IH,13+IV)
,A,PSET
150 IFIH=126 THEN GOTO 170
160 GOTO110
170 PUT(5+IH,5+IV)-(22+IH,13+IV)
,A,PSET:FORX=1TO1000:NEXTX
180 GOTO180
```

Program Listing 2. Draw Space Shuttle, without Flicker

Save London

April 1942. Hitler has unleashed his air weapons against London. As Commander of the English Air Guard, it is your duty to stop the onslaught and save London from certain destruction.

Save London has three phases. In phase one Nazi silent bombers attack at midnight. You are posted at the outskirts of the city with full view of the bombers. You command your three artillery positions hidden in the city by rapidly pressing keys 1, 2, or 3.

In phase two Hilter's secret V2 rockets fly across the afternoon sky, heading for London. You must show leadership by manning an artillery gun and disabling the rockets. Use the right joystick and aim for the rear. Press the F key to fire. The program goes into slow motion when you fire so you can see how close you are shooting.

In phase three you are in a small bomber and sneak behind enemy lines before midnight. Your mission is to drop your payload on Nazi munition plants that make the bombers and rockets. Press B on the keyboards to release your payload.

Each day consists of these three phases. You score points by hitting the silent bombers, disabling the V2 rockets, and hitting the munition plants. The London Blitz only lasts a few days in this scenario, but it's long enough to keep you busy. Before entering, clear memory by turning the CoCo off, then on. If you want to experiment and change the program to suit your own tastes, Table 1 will be useful. ■

Address correspondence to Gary Thomas Wick, 1434 Rutledge St., Madison, WI 53703.

System Requirements

16K RAM
Extended Color Basic
Joystick

```
0 'listing 3  SOUND DEMO. TURN
VOLUME UP
10 FORX=1TO200
20 PLAY "L100;T200;A;C;E;O1"
30 NEXTX
```

Program Listing 3. Sound Demonstration Using PLAY

Program Listing 4. Save London

```
0 '****SAVE LONDON***
10 'BY GARY WICK OF MADISON,WI
20 'COPYRIGHT JUNE '83
30 DIM A(5,5),B(5,5),C(5,5),D(5,
5),E(5,5),F(5,5),I(5,5),H(15,15)
,K(5),L(5,5)
40 DA=DA+1:IF DA>3 THEN GOTO1710
50 CLS(8)
60 PRINT@260,"DAY "DA" OF LONDON
 BLITZ.";
70 FORX=1TO1000:NEXTX:PRINT
80 CLS(0):PRINT"PHASE ONE:":PRIN
T:PRINT"NIGHT BOMBING OF LONDON"
:PRINT:PRINT"USE 1 2 OR 3 KEY TO
 FIRE.":FORX=1TO25:PLAY"O4;L5;T1
30;V30;A;A#;B;C;C#;D;D#;E;F;F#;G
;G#;P3;":NEXTX
90 PMODE 4,1: PCLS: SCREEN 1,1
100 A$="R8;U2;L7;H2;D4"
110 DRAW "BM12,17;XA$;"
120 DRAW"BM22,32;XA$;"
130 DRAW"BM42,46;XA$;"
140 DRAW "BM62,65;XA$;"
150 DRAW "BM42,86;XA$;"
160 DRAW "BM24,100;XA$;"
170 DRAW"BM12,118;XA$;"
180 DRAW"BM0,180;R2U2R2D1E3R2U2R
4D2R2U2D2R2U2R6D6R4U2R2F3R2E4R3U
5R2E4F4D4R8D2R4E4U4R8D4R6U2H4R6U
2R8D6R4D2R6U8R8D8R8E4F4R5E4F4R5E
4F4R3U8E4R5D2R3U3R4D4E4R3U2R5D3R
4U3R5D5R4U4R4FR4D3R5D3R4U3R5D6R8
U3R5D3R4F3R10D3R8"
190 'PLANES IN ARRAYS
200 GET(2,2)-(31,19),A,G
210 GET(16,20)-(32,32),C,G
220 GET(32,32)-(52,56),D,G
230 GET(42,42)-(74,74),E,G
240 GET(32,76)-(52,96),F,G
250 GET(10,90)-(34,110),H,G
260 GET(5,112)-(32,128),I,G
270 'FLY
280 T=0
290 FOR T=1TO300STEP5
300 PUT(2+T,2)-(31+T,19),A,PSET
310 PUT(16+T,22)-(32+T,36),C,PSE
T
320 PUT(32+T,32)-(52+T,56),D,PSE
T
330 IFT>100THENGOTO350
340 F$=INKEY$:IFF$="1" THENGOTO5
10
350 IFT>140THENGOTO370
360 IFF$="2" THENGOTO550
370 PUT(42+T,42)-(74+T,74),E,PSE
T
380 IF INT(T/5)>35 THEN PUT(42+T
,42)-(74+T,74),B,PSET
390 PUT(32+T,76)-(52+T,96),F,PSE
T
400 PUT(8+T,90)-(32+T,110),H,PSE
T
410 F$=INKEY$
420 IFF$="3" THENGOTO590
430 PUT(5+T,112)-(32+T,128),I,PS
ET
440 IF INT(T/5)=36 THENGOTO480
450 V=RND(5):IFV<>3THENGOTO470
460 V=188-RND(5):CIRCLE(T,V),3,1
:PLAY"V31;O1;L2;T80;G;E;C;A"
470 NEXTT
480 CLS:PRINT"END OF NIGHT BOMBI
NG.":PRINT
490 PRINT:PRINT"TOTAL SCORE= "SC
500 FORX=1TO2000:NEXTX:GOTO630
510 H=RND(10)+50:V=RND(100)+30:C
IRCLE(H,V),2:CIRCLE(H+3,V-5),2:P
LAY"L4;T180;O1;C;B;A;G;V30"
520 IF PPOINT(H,V)=5 THEN GOTO 5
30 ELSE 360
530 PLAY"L2;T150;O1;G;F;E;D;V31"
:SC=SC+100
540 GOTO360
550 H=RND(10)+110:V=RND(100)+30:
CIRCLE(H,V),2:CIRCLE(H-2,V-4),2:
PLAY"O1;L4;V20;T180;C;B;A;G"
560 IF PPOINT(H,V)=5 THEN GOTO57
0 ELSE 370
570 PLAY"L2;T150;O1;G;F;E;D;V31"
:SC=SC+100
580 GOTO370
590 H=RND(20)+180:V=RND(100)+30:
CIRCLE(H,V),2:CIRCLE(H+2,V-6),2:
PLAY"O1;L4;V20;T180;C;B;A;G"
600 IF PPOINT(H,V)=5 THEN GOTO61
0 ELSE GOTO 430
610 PLAY"L2;T150;O1;G;F;E;D;V31"
:SC=SC+100
620 GOTO430
630 CLS(0):PRINT"PHASE TWO:":PRI
NT:PRINT"DAYTIME ARTILLARY FIGHT
ING.":PRINT:PRINT"USE RIGHT JOYS
TICK AND F KEY.":FOR X=1TO20:PLA
Y"O4;L4;T150;V30;A;A#;B;C;C#;D;D
#;E;F;F#;G;G#;P5;":NEXTX:PRINT:P
RINT" AIM AT THE REAR TAIL EXHAU
ST    VENTS!";
640 PMODE3,1:PCLS:COLOR2,1
650 V1=0
660 LINE(0,188)-(0,120),PSET
670 LINE(0,120)-(70,120),PSET
680 LINE(70,120)-(70,150),PSET
690 LINE(70,150)-(184,150),PSET
700 LINE(184,150)-(184,120),PSET
710 LINE(184,120)-(254,120),PSET
720 LINE(254,120)-(254,188),PSET
730 PAINT(2,18),7,2
740 K$="R4;G1;E2;R2;U1;L6;H2;D2;
"
750 DRAW "BM10,80;XK$;"
760 SCREEN1,0
770 GET(4,72)-(16,88),K,G
780 GET(4,4)-(14,14),L,G
790 FOR AR=1TO3
800 T=RND(3)-2
810 FORH1=1TO240STEP3
820 PUT(4+H1,72+V1)-(16+H1,84+V1
),K,PSET
830 HJ=JOYSTK(0)
840 VJ=JOYSTK(1)
850 HJ=4*HJ:VJ=4*VJ
860 IF HJ>174 THENHJ=174
870 IFHJ<80 THENHJ=80
880 IF VJ>110 THENVJ=110
890 IFVJ<3 THENVJ=3
900 F$=INKEY$
910 IF F$="F"THEN GOTO1020
920 V1=V1+(T)
930 IF H1=>240THENGOTO990
940 IF V1>30 THENV1=30
950 IF V1<-70 THENV1=-70
960 NEXTH1
970 PUT(4+H1,72+V1)-(16+H1,81+V1
),L,PSET
980 NEXTAR
990 CLS(6):PRINT"END OF DAYTIME
ATTACK.";
1000 PRINT:PRINT"TOTAL SCORE= "S
C;
1010 FORX=1TO1000:NEXTX:GOTO1150
1020 CIRCLE(HJ,VJ),3
1030 CIRCLE(HJ,VJ),3,3
1040 PLAY"O1;L4;T100;V30;D;E;F;G
"
1050 H2=H1+1:V2=V1+1
1060 IF HJ=H2 OR VJ=V2 THEN GOTO
1080
1070 GOTO920
1080 PLAY"O1;L4;T50;V30;D;F;G;E"
1090 SC=SC+100
1100 FORX=1TO5
1110 SCREEN1,1
1120 SCREEN1,0
1130 NEXTX
1140 GOTO920
1150 CLS(0):PRINT"PHASE THREE:":
PRINT:PRINT"NIGHT TIME BOMBING O
F NAZI":PRINT"MUNITION PLANTS.":
PRINT:PRINT"USE B KEY TO DROP BO
MBS.":FORX3=1TO20:PLAY"O4;L4;T18
0;V30;A;A#;B;C;C#;D;D#;F;F#;G;G#
;P5;":NEXTX3:FORX3=1TO2000:NEXTX
3
1160 PCLS
1170 PMODE2,1:CIRCLE(126,95),115
:PAINT(2,2),1:PAINT (200,2),1:LI
NE(116,95)-(136,95),PSET:LINE(12
6,85)-(126,105),PSET
1180 SCREEN1,0
1190 FORX2=1TO3
1200 H1=RND(50)+100:V1=RND(15)
1210 FOR X=1TO62
1220 PSET(H1,V1,1)
1230 PSET(H1+4,V1-1,1)
1240 PSET(H1-3,V1+3,1)
1250 PSET(H1-3,V1+5,1)
1260 PSET(H1+10,V1+3,1)
1270 PSET(H1-15,V1+10,1)
1280 PSET(H1+20,V1+15,1)
1290 LINE(H1,V1)-(H1+3,V1+2),PSE
T,BF
1300 LINE(H1-10,V1+10)-(H1-3,V1+
14),PSET,BF
1310 F$=INKEY$
1320 IFF$="B"THENGOTO1480
1330 LINE(116,95)-(136,95),PSET
1340 LINE(126,85)-(126,105),PSET
1350 PRESET(H1+4,V1-1)
1360 PRESET(H1-3,V1+3)
1370 PRESET(H1-3,V1+5)
1380 PRESET(H1+10,V1+3)
1390 PRESET(H1-15,V1+10)
1400 PRESET(H1+20,V1+15)
1410 PRESET(H1,V1)
1420 LINE(H1,V1)-(H1+3,V1+2),PRE
SET,BF
1430 LINE(H1-10,V1+10)-(H1-3,V1+
14),PRESET,BF
1440 V1=V1+3
1450 NEXTX
1460 NEXTX2
1470 CLS(4):PRINT"LOW ON FUEL.":
PRINT"MUST FLY BACK TO BASE.":FO
RX=1TO2000:NEXTX:GOTO40
1480 FORT=1TO30
1490 H=126:V=105
1500 X=RND(3)
1510 IFX=1 THENH=H-10
1520 IFX=2 THEN H=H+9
1530 IFX=3 THENH=H
1540 X=RND(3)
1550 IFX=1 THENV=V+10
1560 IFX=2 THENV=V+15
1570 IFX=3 THENV=V
1580 B1=RND(4):V=V+B1:H=H+B1
1590 CIRCLE(H,V),4,1
1600 IFH=H1 THENSC=SC+30
1610 IFV=V1 THENSC=SC+150
1620 IFV=X THENSC=SC+200
1630 SCREEN1,1
1640 SCREEN1,0
1650 PLAY"O1;L1;T150;V15;C;G;F;"
1660 NEXTT
1670 CLS(0):
1680 PRINT:PRINT"TOTAL SCORE= "S
C;
1690 FORX3=1TO2000:NEXTX3
1700 GOTO 40
1710 CLS(8):PRINT@170,"END OF BL
ITZ";:PRINT@32,"TOTAL SCORE IS "
SC;:END
```

END

Line 30 Dimensions arrays for bombers.
Lines 100–180 Draw bombers and London.
Lines 200–260 Place bombers at start position.
Lines 270–620 Fly bombers and shoot artillery routine.
Line 630 Starts phase two.
Lines 660–720 Draw artillery gun view.
Line 730 Paints sky.
Lines 770–820 Fly V2 rockets.
Lines 830–890 Set joystick limits.
Lines 900–1140 Fire artillery.
Line 1150 Starts phase three.
Lines 1180–1450 Draw and move munition plants.
Lines 1480–1690 Drop bombs routine.

Table 1. Save London Line Descriptions

↙ See List of Advertisers on page 99

The Educated Guest

CONSIDER THE HANDICAPPED

by Charles H. Santee

Nothing is more frustrating than having an idea and lacking the means to communicate it. Those of us with such means of expression and reception as sight, speech, hearing, touch, and the ability to write have experienced only minor frustration. We have not only our natural communication abilities but an impressive array of technical advances as well.

Consider the handicapped. How do the communication-impaired deal with these frustrations? How can we develop new forms of communication to free the handicapped and enhance our own normal abilities?

My challenge to you this month is exactly this, and the Color Computer offers an answer. Examine current hardware and software developments and use your own imagination and intelligence to explore new communication ideas and define how existing ideas for the handicapped can be put to use by all of us.

Resources

A surprising number of resources in computer applications for the handicapped exist. You will find a partial list of these organizations in Table 1. I encourage you to write to them for further information.

Preexisting Materials

Much hardware and software designed for the general public can be put to good use by the handicapped. The word processor often provides enough of an edge to make written communication possible. In fact, Elite-Word (by Elite Software, P.O. Box 11224, Pittsburgh, PA 15238) is now being evaluated for specific use with the visually handicapped.

Recorded, digitized, or synthesized speech is good for presenting text to nonreaders. Dorsett Educational Systems' English as a Second Language, Learning Games for Children by DD Software (10 Simonne Lane, Pepperell, MA 01463), The Voice from Speech Systems (38 W 255 Deerpath Road, Batavia, IL 60510), and Spectrum Voice Pak and Term Talk (a talking terminal program) from Spectrum Projects (93-15 86th Drive, P.O. Box 21272, Woodhaven, NY 11421), all offer exciting possibilities for the mentally and physically handicapped.

Equally exciting, but just as undeveloped is the use of peripherals such as joysticks and lightpens as alternatives to the keyboard.

A Place to Start

These are promising applications with plenty of room for improvement for the general public as well as the handicapped. Recorded voice programs such as those offered by Radio Shack, Dorsett, and DD Software present material in a serial format, which allows no computer-controlled branching or random selection of presentation order for spoken material. A mix of serial verbal content and randomly presented text and graphics would help.

The Learning Games program (DD Software) skips certain recorded messages according to student responses and uses an interesting mix of graphics presentations to avoid the effect of a

There seems to be a symphony
that whirls within their minds
And with their every movement
flows symmetrical designs
What is the mystic sight they see
What golden dream, what prophecy
Each silenced soul confines?

Alan J. Brown Center for Alternate Communication and Environmental Control Rehabilitation Institute of Chicago
345 East Superior St.
Chicago, IL 60611
312-649-6000

Communication Outlook
(a quarterly publication focusing on communication aids and techniques)
Published by the Artificial Language Laboratory and The Trace Center
Annual subscription rate $12.

Special Net
(the largest education-oriented computer-based communication network in the United States)
Write the National Association of State Directors of Special Education.

Artificial Language Laboratory
Computer Science Department
Michigan State University
East Lansing, MI

National Association of State Directors of Special Education
1201 16th St. N.W.
Washington, D.C. 20036
202-822-7933

Trace Research and Development Center for the Severely Communicatively Handicapped
University of Wisconsin-Madison
314 Waisman Center
1500 Highland Ave.
Madison, WI 53706
608-262-6966

Table 1. Resource Guide

slide-show presentation. Despite the flaws in this program, young nonhandicapped, and older mentally handicapped students seem to enjoy and learn from it.

In the area of speech, recorded and digitized varieties are good for random presentation, but they take too much memory and leave little room for creative control. Synthesized speech is a better alternative for the handicapped.

Spectrum Projects' Spectrum Voice Pak has potential, but much of the support material now produced for the speech synthesizer needs considerable improvement. Often speech is simply added to an existing program, and more often, it is not added at all. The pronunciation is often flawed, but despite these problems children use the computer much more effectively with some means of synthesized speech.

Word processors for the Color Computer are generally good, but here again, there is room for improvement. I would like to see more word processors designed for children, with screen information in a simpler format and an icon approach to the editing process.

A word processor that uses a "look-up and type-ahead" dictionary could be of real use to the handicapped since it requires fewer keystrokes. In this particular process the user hits a letter key causing the computer to look up the first word in the dictionary with that letter. As he hits the next key, the first word with those two letters appears, and so on. When he sees the desired word, the user hits a control key to capture the word in text.

Adaptations for the Handicapped

Ideally, all software and hardware should be useful in communication for handicapped and nonhandicapped alike. We have made significant advances in this area though we are far from the ideal.

Commonly, the handicapped can have a matrix, board, or dictionary of words or symbols to express what they want to communicate. There is some method of selection including pointing, pressing a key, moving a joystick, puffing on a straw, looking in a given direction, or even moving a single muscle. This method is often highly specific to an individual, expensive, and impractical for use in a commercial product. It is often only partly effective. The computer is our means, at a reasonable cost, to transpose many forms of input into output in a more standardized, practical approach.

At the University of Michigan, Dr. Eulenberg and several of his colleagues have been using the Color Computer, with joysticks and speech synthesizer, to develop communications programs. They have modified the joysticks for output and control as well as input, and use them for such things as controlling television channel selection.

Dr. Eulenberg shares my enthusiasm for the Color Computer in communications since the digital-to-analog converter, easily programmable graphics, tape recorder control, and other features make it well suited for these applications.

An adaptive firmware card for the Apple (developed by Paul Schwdja and Greg Vanderheiden of the University of Washington and The Trace Center) lets you bypass the keyboard and use any device that will hook up to the firmware card. He can use a traditional spreadsheet program through an external touchplate, as well as many other types of software without modification. This device is certainly worth developing for the Color Computer.

Comunicaytions Pleez!

Here is my challenge to you:

- Explore computer applications for the handicapped. Write to the addresses I give to discover your own sources of information. Computer applications for the handicapped have important implications for any software developers.
- Share information with other people. Let schools, friends, and associates know what is going on. You might help a handicapped person develop better communication.
- Consider special adaptations in the development of traditional software and hardware. Here are a few suggestions:

1. Develop a screen-to-speech-synthesizer dump—when you touch a control button, the synthesizer reports what is on the screen. This lets you speak out with software not developed for the synthesizer.
2. Incorporate a screen slow-down routine in your programs.
3. Provide for input in your applications that allows alternatives to a standard keyboard.
4. Make a keyboard emulator.
5. Develop the "picture perfect" word processor.
6. If ideas are all you have, then communicate your ideas. ■

Write to Charles Santee c/o The Educated Guest, HOT CoCo, *80 Pine St., Peterborough, NH 03458.*

Dr. Charles Santee will be appearing at the Chicago Rainbowfest (June 22 and 23) as a participant in the Educator's Forum. Our own "Educated Guest," along with Dr. Michael Plog (Program Evaluator for The Illinois State Board of Education), and Julie McGee (Radio Shack's Director of Software Development), will discuss the issues of evaluating educational software. This is to be a round-robin discussion of methods for choosing and evaluating software, accompanied by a look at past and future trends in educational software.

MAKE IT EASY TO SAVE your copies of

HOT CoCo

Your magazine library is your prime reference source—keep it handy and keep it neat with these strong library shelf boxes. They are made of white corrugated cardboard and are dust resistant. Use them to keep *all* your magazines orderly yet available for constant reference.

Self-sticking labels are available for the following:

80 Micro	*73 Magazine*	*Radio Electronics*
Microcomputing	*QST*	*Personal Computing*
inCider	*CQ*	*HOT CoCo*
RUN	*Ham Radio*	*Interface*

One box (BX1000) is $2.00, 2–7 boxes (BX1001) are $1.50 each, and 8 or more boxes (BX1002) are $1.25 each. Be sure to specify which labels we should send.

Call **TOLL-FREE** for credit card orders:

1-800-258-5473

Or use the order form in this magazine and mail to:

HOT CoCo
Attn: Book Sales, Peterborough, NH 03458

☐ SHIPPING AND HANDLING CHARGES $2.00 per order, up to and including a quantity of eight. 25¢ for each additional box ordered. ☐

HOT CoCo

Reader Service

TO RECEIVE MORE INFORMATION ON THE PRODUCTS AND SERVICES ADVERTISED IN THIS ISSUE, PLEASE TURN TO READER SERVICE CARD.

MUL-T-SCREEN

RAINBOW CERTIFICATION SEAL

COLOR CHARACTER GENERATOR

A NEW DIMENSION IN COLOR COMPUTING

RAINBOW CERTIFICATION SEAL

Now includes a character generator and sample graphic space game at no extra cost.

Full 224 text and graphic characters. Underline in all PMODES. Easily combines text and graphics in any PMODE.

All machine language, user transparent. As easy to use as the standard print screen. Supports all BASIC, EXTENDED BASIC and DISK commands (tape version does not support disk drive).

Automatic loader recognizes 16K, 32K and 64K BASIC and EXTENDED BASIC computers and loads MUL-T-SCREEN accordingly.

Can mix up to 5 character sizes, 4 character colors and 4 background colors all on one screen. A total of 10 character sizes are available from 8*4 in PMODE 0 to, 42*24 in PMODE 4. Will print vertically in PMODE 4 in a 32*32 format.

Includes 3 EXTENDED BASIC Sample Programs (a program for those without EXTENDED BASIC is also supplied).

Up to FOUR defineable limited text (window) screens can limit the text anywhere from a single character to the entire screen. Each limited screen can have different character size as well as different character and background colors. Very easy access from one window to another.

Includes positive and negative screen dumps in two sizes. (Gemini and Epson versions also available - please specify)

Horizontally scrolling (crawling) one line screens. Up to FOUR can be used at the same time.

Special Trace Delay can be used for debugging graphic programs. While in operation the line number appears in one corner of the screen while the graphic result of the line appears on the remainder. A key needs to be pressed before the execution of each line.

A special printer control can output characters to the screen and printer simultaneously.

MUL-T-SCREEN is a must for all color computer owners. The large color characters are prefered by children and make learning a lot more fun. Invaluable to all serious programmers. MUL-T-SCREEN can handle almost any task imaginable. Once you try it, you won't write another program without it.

INCENTIVE SOFTWARE

(519) 681-0133

P.O. BOX 323
STATION B
LONDON ONTARIO
CANADA N6A 4W1

P.O. BOX 7281
PORT HURON
MICHIGAN 48301
U.S.A.

VISA

MINIMUM REQUIREMENT 16K BASIC
TAPE - 24.95 US or 29.95 CDN
DISK - 27.95 US or 32.95 CDN

MasterCard

TAPE TO DISK UPGRADE AVAILABLE FOR $8 US OR $10 CDN. OUTSIDE US AND CANADA ADD $2. ONTARIO RESIDENTS PLEASE ADD SALES TAX. WE PAY ALL SHIPPING AND HANDLING. VISA AND MASTERCARD CHARGED IN CANADIAN FUNDS PLUS $1.

91

Dealers Dealers Dealers SELL!

Selling **HOT CoCo** will make money for you. Consider the facts:
Fact 1: Selling **HOT CoCo** increases store traffic—our dealers tell us that **HOT CoCo** is one of the hottest-selling computer magazines on the newsstands.
Fact 2: There is a direct correlation between store traffic and sales—increase the number of people coming through your door and you'll increase sales.
Fact 3: Fact 1 + Fact 2 = INCREASED SALES, which means more money for you. And that's a fact.

For information on selling **HOT CoCo**, call 800-343-0728 (In N.H. call 924-9471) and speak with Ginnie Boudrieau, our bulk sales manager. Or write to her at **HOT CoCo**, 80 Pine Street, Peterborough, NH 03458.

HOT CoCo

80 Pine Street
Peterborough, NH 03458
800-343-0728

FREE OFFER!

COMPUTER CASSETTES 58¢

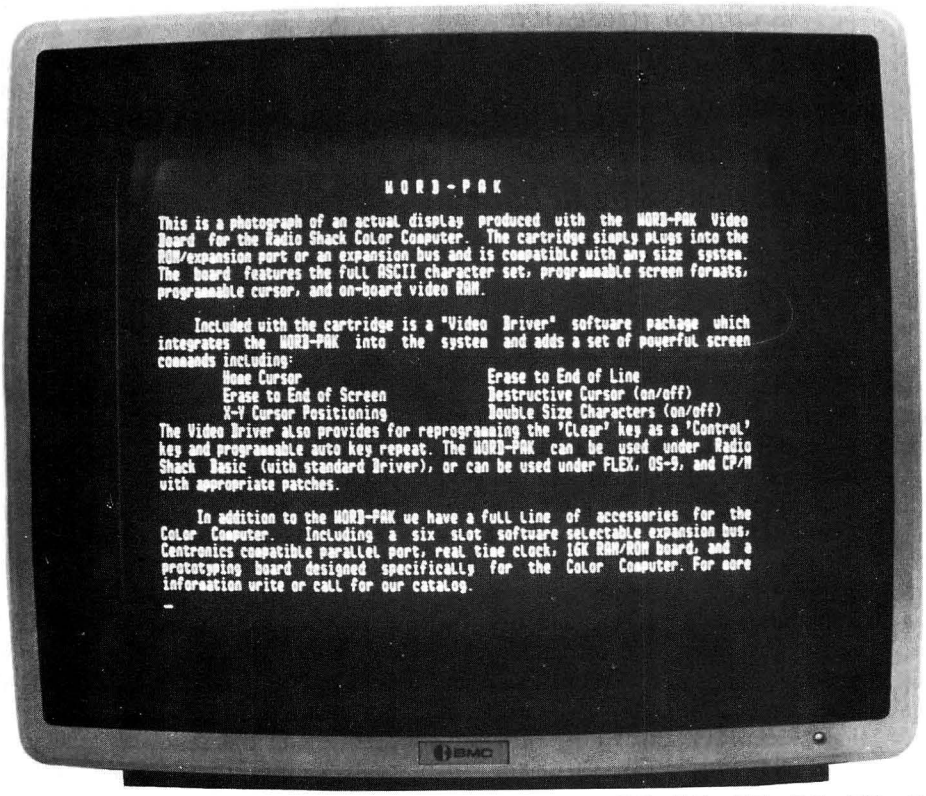
FREE "World Capitals Game Cassette"
with each order of 20 or more C-10's
Specify TRS-80 Color Computer, MC-10, TI-99/4A VIC 20 or Commodore 64

- **C-10 Length**
- **5 Screw Shell / Free Labels**
- **Lifetime money back guarantee**
- **Storage Box add 12¢ each**
- **$2.00 shipping charge - any quantity (Canadian Orders $6.00 shipping)**
- **NJ Residents add 6% Sales Tax.**
- **Send check or money order to:**

PARALLEL SYSTEMS ↙207

Box 772 - Dept. H
Blackwood, NJ 08012
609-227-9634

A Picture Is Worth . . .



PJ inc. ↙214

P.O. BOX 813
N. Bergen, N.J. 07047
(201) 330-1898

↙ See List of Advertisers on page 99

The DOSsier

THE COCO GETS DOWN TO BUSINESS

by Scott L. Norman

Well, I've finally done it. After all my claims that the CoCo is a good small-business machine, I've put my money where my mouth is. In addition to my elderly, silver machine at home, I now have a hot new, white 64K model in the office. I fully expect it to become my principal testbed for Flex and OS-9 applications software.

I had used an Osborne 1 in my office. I loved SuperCalc, disliked WordStar, and grew frustrated with the Milestone project-management package. Early this year, I had grown weary enough of software incompatibility and multiple operating systems to ditch the Osborne—CP/M, WordStar, and all. I had run enough applications software through my CoCo to know that it could handle most of my professional requirements, so I made it my professional machine.

The 64K CoCo came at an end-of-production sale price, but I outfitted it rather handsomely. I opened the case and installed one of Computerware's little Video Plus driver boards (P.O. Box 668, Encinitas, CA 92024, 619-436-3512), so that I could use the 13-inch Zenith monitor left over from my Osborne days. I also dropped in an HJL keyboard (955 Buffalo Road, P.O. Box 24954, Rochester, NY 14624, 716-235-8358). The stock keyboards on the new CoCos look sporty, but they really don't work out for word processing. Chiclets are chiclets, no matter what kind of keycaps you put on them.

As for disk drives, I needed maximum reliability and wanted to try out some new technology at the same time, so I splurged and went for a J&M System Ltd. (137 Utah NE, Albuquerque, NM 87108, 505-265-1501) controller and a pair of half-height TEAC drives, once again from Computerware. The Frank Hogg Laboratory, (The Regency Tower, Suite 215, 770 James St., Syracuse, NY 13203, 315-474-7856) and other vendors carry comparable equipment, by the way.

They work beautifully. The speed test on the OS-9 boot disk says that my drives are turning over at 299.5 rpm, and they look as though they're prepared to stay there forever. J&M's phase-lock-loop technology has a lot to do with that, I'm sure.

At this point, I topped things off with a Botek (4949 Hampshire, Utica, MI 48087, 313-739-2910) interface box for my Epson FX-80 printer, and I had one high-class system. The Video Plus turned out to be a trifle low on output, even after I adjusted its trim potentiometer, but Flex's regular 51-by-24 screen format still looked fine on the monitor. I soon had Stylograph, DynaCalc, and some of my favorite RSDOS math and graphics programs churning away.

"An 80-column display is almost irresistible for us spread-sheet addicts, and it can be awfully useful for word processing to boot. I went for it."

Not long after I began to use this setup, however, an intriguing new product appeared: Word-Pak, an 80-column, 25-line display board from PBJ Inc. (P.O. Box 813, North Bergen, NJ 07047, 201-330-1898). An 80-column display is almost irresistible for us spreadsheet addicts, and it can be awfully useful for word processing to boot. I went for it.

The process of installing the driver software so that my applications programs could capitalize on the new display turned out to be pretty instructive, so I'd like to describe it in some detail.

A Display Just Like the Big Guys'

Word-Pak is a video-driver board that plugs into the CoCo's cartridge slot. It's packaged in a case like the Radio Shack disk controller, and you can use the two simultaneously by hooking them up to a Y-cable or an expansion interface. A nice, clean baseband video signal emerges from an RCA phono jack at one end; if you haven't guessed by now, this is strictly a gadget for the video-monitor crowd.

That baseband signal is potent, by the way. I have to turn the Black Level—or gain control—on my monitor almost all the way down to get a clean display.

Word-Pak comes with a cassette containing drivers for both cassette and RSDOS disk systems. To use it with Flex (Frank Hogg Lab version only), you need the Flex Patch, a separate $29.95 program. The patch disk holds four files for each of the four current versions of FHL Flex, numbers 5.0:1–5.0:4.

The first of these four files operates just like Flex's regular high-resolution screen driver, and lets you switch back and forth between the 51-by-24 and 80-by-25 displays. You can append the second file to Flex to make the system wake up in the higher-density mode. The last two files contain the source code for each .CMD and .BIN routine.

If you decide to go with the .CMD command, the Word-Pak driver will load into the same portion of RAM normally used by the Flex driver and perform all the necessary patches to route output through the outboard

Unleash the power of your computer!

You've heard that line before, right? And you know that it really means "buy my program, and good luck to you."

Well, this ad is for people who don't mind a little hard work. I'm talking to those of you who believe that owning a computer means a lot more than buying somebody's expensive software or blindly typing in magazine programs — with the fervent hope that either one will work as promised.

Am I talking to you? If so, then "Learning the 6809" should interest you. For $99 plus shipping, I'll send you my complete course in how to program the 6809 processor, the powerful heart of your Color Computer. It's just about the same course I teach in our state college system, and a better bargain because I'm always there on tape if you need to consult me.

Now listen. If you weren't born to program, then "Learning the 6809" won't do you any good. But if you've got the inclination to program and a willingness to work, then the Micro Language Lab is what you're looking for.

My 1,500-question programmed learning system is enough to teach you everything the pros know, but that's just a small part of the course. I've got 24 half-hour lessons on cassette, a 230-page book, 100 pages of data, and 35 programs that will let you — not someone else's software — "unleash the power of your computer."

If you're skeptical, write to Green Mountain Micro. I'll send you a copy of the table of contents and a sample page from the workbook. Or if you're ready to get started right away, order your Micro Language Lab by phone or mail. Requires 16K extended basic and EDTASM+.

What is the ultimate all-software music synthesizer for your TRS-80 Color Computer?
What is the least expensive all-software music synthesizer for your Color Computer?
Quaver, the most realistic music experience you'll ever hear from your Color Computer!

COLOR QUAVER

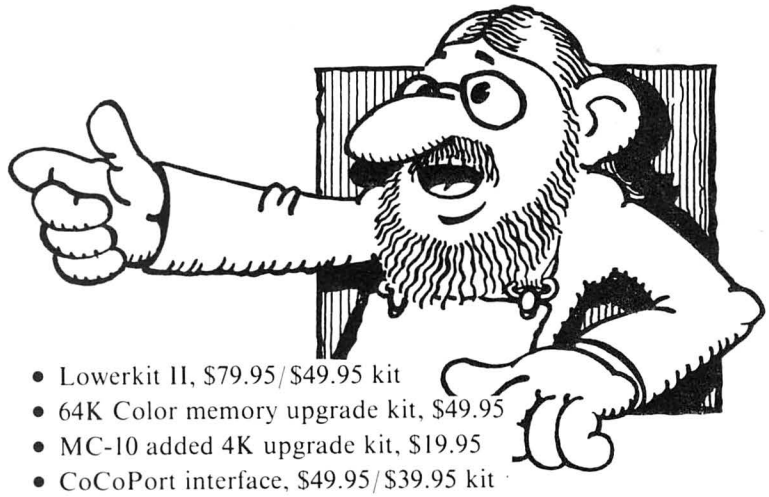
Software Music Synthesizer

(New Version 2.1 for 32K ECB)

by Dennis Bathory Kitsz author of "Custom Color"

- Real Music Synthesis — More Than Bleeps!
- Full 4-Part Harmony — In Precise Tuning.
- Versatile Editor/Compiler/Storage System.
- Entirely Software — No Hardware Needed.
- Variable Tone Qualities for Each Part.
- 64-Step Variable Envelope for Each Part
- FAST Compiler — Finished Music in 5 Seconds!
- Ready-to-Play Sample Tunes Included.

— Special Price —
Only $19.95 on tape
(Add $2.50 shipping and handling)
(Not sure? Audition Our Sample Music Tape — $4.00 postpaid)
For 32/64K TRS-80 Color Computers with Extended Color Basic.



- Lowerkit II, $79.95/$49.95 kit
- 64K Color memory upgrade kit, $49.95
- MC-10 added 4K upgrade kit, $19.95
- CoCoPort interface, $49.95/$39.95 kit
- RAM/ROM pack, $29.95/$19.95 kit
- Micro Language Lab — "Learning the 6809" — $99/$3.50 S&H
- Color Burner assembled & tested with software $69.95/$57.00 kit with software
- Scroll-A-Roll software video text display, $24.95
- TV Buff II, improved to handle virtually all monitors, $14.95

(Add $2.50 shipping and handling — Canadian orders add $1.50 more)

Green Mountain Micro

Bathory Road • Roxbury, Vermont 05669
(802) 485-6112

Hours: Mon–Fri. 9-5

COD/VISA/MASTERCARD

Dealer Inquiries Invited.

TRS-80 is a trademark of Tandy Corporation

See List of Advertisers on page 99

cartridge. This disables the TV set or monitor connected to the CoCo, and the monitor connected to the Word-Pak comes alive. If you're using a single monitor, either build a little switch box or else resign yourself to doing a lot of cable swapping.

If you append the .BIN file to Flex to create an auto-starting, high-resolution version, you gain about 2K of RAM—the area above $B800 otherwise used by the Flex screen drivers, the manual says. The only disadvantage is that you won't be able to use the DBASIC or EXT commands, both of which try to copy that portion of memory into the section now occupied by the Word-Pak driver. You might then expect resounding system crashes, but somehow I've never been able to make myself care.

The procedure for appending the file to Flex, as opposed to incorporating it into an applications program, is pretty simple. Format a new disk, put the boot loader on it, copy the appropriate display driver, append the driver to Flex to create a new operating system, and link this to the loader.

I use Flex 5.0:3, so if my master disk is in drive 0 and I'm building a new version called FLEX80 in drive 1, the last two commands are as follows:

```
APPEND,0.FLEX.SYS,1.X80PAT3.BIN,1
.FLEX80.SYS
LINK,1.FLEX80.SYS
```

The only thing that remains is to copy any utility commands I want onto the new disk. The RUN "FLEX" command then brings up FLEX80 and the 80-column display immediately.

Doing the Deed with DynaCalc

Word-Pak generates a classy-looking display with all sorts of cursor control functions, but remember that my primary purpose was to use it for applications software—not programming. This requires some work, because most applications programs that run under CoCo Flex have undergone a certain amount of conversion already: They were originally written for other terminals with 80-column displays, and their authors had to go to some trouble to squeeze them onto a 51-by-24 screen. Now the idea is to undo such efforts.

My first target was DynaCalc, the spreadsheet from Computer Systems Center (13461 Olive Blvd., Chesterfield, MO 63017, 314-576-5020). I reviewed it at length in the October 1983 issue of this magazine (p. 24), so I'll just point out that it's a pretty powerful item, with almost all the features of any 8-bit spreadsheet I can think of. I use the program frequently, and thought the difference between 51 columns and 80 was worth a little effort.

"Lacking the utilities to change the critical locations right on the disk . . . I took the honorable way out: I called for help."

The general approach is simple enough: Change the portion of the applications program that defines the screen format and then save the revised code to disk. The folks at PBJ supposedly furnish a sheet that describes the locations to be changed in DynaCalc, Stylograph, DynaStar, and the RMS database manager, but I found no such sheet in my package. However, a phone call soon corrected that.

Getting the straight story on the range of addresses to be saved took a little more doing, primarily because my kit of Flex utilities was somewhat understocked. I've since taken care of that, but I think it's worthwhile to go through the exercise the way I had to do it.

DynaCalc stores the number of display rows in location $010D, and the number of columns, minus one, in $010E (this according to PBJ). The numbers themselves are in hex, so the change from 51-by-24 to 80-by-25 requires that you change the contents of $010D from $18 to $19, and those of $010E from $32 to $4F.

There are utilities that let you carry out such alterations right on the disk, but since I didn't have one at the time I did this job, I used the monitor routine included with FHL Flex to change the locations after DynaCalc was in RAM.

This left me with the problem of finding the program's location, so I could use Flex's SAVE command to store the modified version. By using the monitor to examine low memory both with and without DynaCalc aboard, I convinced myself that the spreadsheet started to load at $0000—but how high did it go?

The disk catalog revealed that the original FLEX.SYS file occupied 108 sectors, which meant that there might be as many as 108×252—or 27,216 bytes—of program in there (machine-language files have a 4-byte header and 252 data bytes per sector). This implied that I should save everything up to $6A50, the hex equivalent of 27,216. Of course, I had no way of knowing if the program used all 252 bytes in every sector.

Worse yet, I had no idea if DynaCalc actually occupied one contiguous chunk of RAM. Lacking the utilities to change the critical locations right on the disk, and being unable to examine the sector headers directly, I took the honorable way out: I called for help. Fortunately, Joe Turner at Computer Systems Center was willing to set me straight, and to be quoted.

It turns out that DynaCalc uses a big slug of memory starting at $0000, all right (that's the part that includes the changes for Word-Pak), but there is also a piece that resides up in the utility-command space starting at $C100.

The simplest thing for me to do, once I knew the extent of each piece, was to save them separately and then append them to form my new version of DynaCalc. This came about because the SAVE command itself would normally be located in the utility-command space, so I had to use SAVE.LOW to handle the part of DynaCalc that lives there: this is the standard way of saving programs from that portion of RAM.

Here, then, in gory detail, is how to do the entire job, starting with DynaCalc on the master disk in Drive 0. Things following a + + + sequence are Flex command lines, of course, and I've been more explicit with drive identifiers than I really had to be.

- + + +GET,0.DYNACALC.CMD—Remember, GET loads a binary file into memory without executing it. It is part of the operating system, and not a utility command.
- + + +SAVE.LOW,1.DYNA.HI,C100,C684—This puts the high-mem-

ory portion of DynaCalc onto disk. Joe Turner supplied the start and finish addresses; no transfer address is necessary, as it will come with the other part of DynaCalc. Appending the two parts preserves it.

- Using the FHL monitor, I filled memory from $0000–$65F1 with zeros; this may not be absolutely necessary, but it gets rid of SAVE.LOW and anything else that may interfere with the rest of the process. Execute a warm jump back to Flex when you're finished.
- + + + GET0.DYNACALC.CMD—One more time.
- Use the monitor to change $010D and $010E, then execute another warm jump back to Flex.
- + + + SAVE,1.DYNA.LO,0000,65F1,0000—Now both parts of DynaCalc are on disk.
- + + + APPEND,1.DYNA.LO,1.DYNA.HI,1.DYNA80.CMD—And now you have made one file out of them.

A little housekeeping finishes the job. I deleted DYNA.LO and DYNA.HI from my new working disk and used the BUILD command to set up the two files that cause the system to wake up in the new spreadsheet, which I have called DYNA80:

```
GO.TXT:
        SETUP PB9600,R
        P DYNA80
STARTUP.TXT:
        EXEC GO
```

The P command is necessary to specify the printer driver when you first call DynaCalc.

Yes, it was worth the trouble, although next time I'll use one of the Flex utilities to do the job more neatly. The new 80-column format is a significant improvement, and lets me see the entire width of some of my most frequently used sheets at one time. The character set is nice too, although it is proportionally somewhat taller than what I'm accustomed to.

To top it off, getting rid of Flex's regular screen drivers has boosted DynaCalc's empty-sheet capacity by more than 1.5K, from 22,396 bytes to 23,944, without getting rid of the Help screens. ■

Address correspondence to Scott Norman, 8 Doris Road, Framingham, MA 01701.

↙ See List of Advertisers on page 99

Doctor ASCII

by Richard E. Esposito and Ralph E. Ramhoff

Q. I purchased a Radio Shack Screen Print Routine #26-3021 for my 32K CoCo and DMP-120, but whenever I try to print anything from the screen, I get garbled graphic characters on the left edge of the page. What is wrong?

Edward E. Hott
Marion, OH

A. You need the new version, which is known as BW-dump, and it is contained in Radio Shack's Hi-Res Screen Print Utilities, #26-3121, $9.95. The version that you have was written for the older LP-VII and LP-VIII printers.

Q. I recently purchased a 64K Extended Basic CoCo with a DMP-100 printer, Color Scripsit, and other accessories but no drive. My problem is that when using Color Scripsit and typing with upper- and lowercase letters, all lowercase letters that would normally fall below the line such as p and y are raised up onto the line when printed. Does the problem lie in the computer, the printer, or in the ROM pack, and how can I correct this?

Louis Pereira
Fall River, MA

A. I'm sorry to be the bearer of bad news, but the problem lies in your selection of a printer. The DMP-100 lacks what are known as "lowercase descenders." You can write a program that will give you a different character set using the graphics mode of your printer, but this would rule out its use with most word processors. A hardware solution would be to replace the printer's internal ROM with a custom EPROM containing a revised character set. If anyone has done this, please let me know and I'll mention it here. Before buying a printer, it is a good idea to look at a sample of the complete character set that it prints so that you will not be disappointed later.

Q. The Extended Basic manual talks about built-in machine-language routines that are available. I have tried to use JOYIN and POLCAT to no avail and have gotten no help from Radio Shack.

Lynn H. Sundberg
Jacksonville, FL

A. I assume that you are attempting to call these routines from a machine- (Assembly-) language program. There is no easy way to use these routines from Basic; besides, the INKEY$ and JOYSTK functions provide the same facilities. The Radio Shack manual shows the addresses of JOYIN and POLCAT as $A00A and $A001, respectively. The manual, however, fails to tell you that these are not the addresses of the routines but only the addresses of the address pointers. In other words, the 2 bytes at $A00A and $A00B together contain the address of the JOYIN routine transfer address.

To call the JOYIN routine you must JSR to the routine whose address is contained in bytes $A00A and $A00B. This can be accomplished by the use of indirect addressing (i.e., JSR [$A00A]). Notice the square brackets; this is how indirect addressing is indicated in Assembly language. The directions in the manual give the information you need to use the subroutine after you figure out how to call it. The following program segment illustrates the use of the JOYIN routine to read the joystick values into a user area. After reading the joystick values it calls POLCAT to read a character into the A accumulator.

```
                .
                .       (your program starts here)
                .
                LDX     #????           Put the JOYSTK values here
                JSR     JOYSTK          Go read the Joysticks
                .
                .       (your program continues here)
                .
                JSR     KEYBRD          Go look for a keystroke
                .
                .       (the rest of your program is here)
                .
JOYSTK          PSHS    A,B,X,Y,U,CC    Save the registers
                JSR     [$A00A]         Call JOYIN
                PULS    A,B,X,Y,U,CC    Restore the registers
                PSHS    A,B,Y           Save D and Y
                LDY     #$015A          Start addr of JS values
                LDD     0,Y             Load left JS values
                STD     0,X             Store at 0,X and 1,X
                LDD     2,Y             Load right JS values
                STD     2,X             Store at 2,X and 3,X
                PULS    A,B,Y           Restore D and Y
                RTS                     Return from subroutine
KEYBRD          PSHS    Y,U             Save the registers
                JSR     [$A001]         Call POLCAT
                PULS    Y,U             Restore the registers
                RTS                     Return from subroutine
```

Note: Save all the registers before you call the ROM subroutines, because the ROM routines destroy the contents of the registers.

Q. My Gemini 15 printer's instruction manual is very limited, so I do not have any idea how to send control codes to my printer from a Basic program. I have used my printer with Elite-Calc, so I know the printer is working properly.

R. Renalds Neely
Asheboro, NC

A. The Gemini printers come standard with a minimal instruction manual. When I bought my printer I had to ask for the expanded instruction guide. Fortunately, I had

been informed that there was a better manual available. This upgraded manual is spiral bound. Your manual gives you a list of control codes with both their hexadecimal and decimal values. To transmit these codes to the printer you need to use the decimal value(s) in CHR$ function(s) in a PRINT#−2 statement. The following program will ring the printer's "bell" four times:

```
10 FOR I = 1 TO 4
20 PRINT #-2,CHR$(7);
30 NEXT I
```

Q. I have an Okidata Microline 92 printer and a 64K cassette-based CoCo. I need a program that will copy a graphics page to my printer. Can you help me?

Mark Smith
Greenville, TX

A. I do not have access to a Microline 92, but Custom Software Engineering Inc., 807 Minuteman Causeway (D-2), Cocoa Beach, FL 32921, markets one for $9.95, plus $1 shipping. To my knowledge, they have the most extensive catalog of screen-dump programs for dot-matrix printers.

Q. Is there a program to convert a Model III program for use on the Color Computer? There are some programs that I would like to purchase that are only available for the Model III. I have a 64K CoCo with a Tandon disk drive and a relative that has a Model III with two disk drives, and this conversion would be a great help.

Esther Horst
Sterling, OH

A. There are two programs. One, marketed by Spectrum Projects (93-15 86th Drive, Woodhaven, NY 11421), gives you a 64-by-16 screen with Model I/III graphics symbols. The program is called 64-Column Mod I/III Emulator and sells for $19.95, plus $3 shipping. The other program, marketed by MichTron (1691 Eason, Pontiac, MI 48054), runs on a Model I/III/4 machine allowing you to transfer files to or from Color Computer disks. This program is called CIII and sells for $24.95, plus $3 shipping. Remember that even though the data and programs can be successfully transferred, they may not execute properly on the Color Computer. If the program is written in machine language, forget it. If the program is written in Basic and it is in ASCII format, you may need to make some changes to get the program to run. This is especially true if it includes PEEKs and POKEs.

Q. I recently bought Radio Shack's Madness and the Minotaur (#26-3313) adventure game and I'm getting nowhere with it. Would you please give me some information concerning the proper play of the game, or at least give me an address where I can write for help?

Edward G. Nowak
Fredonia, NY

A. Perhaps a few adventure fundamentals will help you out. All adventures take place in an adventure world. In these worlds, anything can happen, and it usually will. The first step in solving any adventure is to construct a map of its world. You do this by recording your movements on paper. Every time you move from one room to another, place the new room on the map and connect it to the room that you just left.

In order to move around the adventure world, you first need to know the vocabulary of the adventure program. In the March 1984 Doctor ASCII (p. 135) there was an adventure-peeking routine. This program will help you determine all the words that the program understands.

Never assume that anything in an adventure is insignificant! Until you are told that the chair is "not interesting," don't ignore it. You must look at, around, or into everything. For example: When you come upon a pool of water, you can drink it, swim in it, look for a reflection in it, or even fill a bottle from it. The program Madness and the Minotaur was originally written by Spectral Associates (3416 South 90th St., Tacoma, WA 98409) and they market a help sheet for a nominal fee.

Q. My problem seems to occur when using large string arrays with my 32K Extended Basic Color Computer, and printing them on the screen, although the quantity printed seems to have little effect.

From "Journey to the Center of the ROM—Part IV," *HOT CoCo*, February 1984, p. 102, I note that as the program is running the address pointed to by $25, $26 approaches the address pointed to by $17, $18. When they almost match, my machine will stop execution for up to a minute and a half. As a wild guess, it seems that the CoCo has just taken all the string information and repacked it, putting it back into a smaller area, before continuing. This seems impossible, so what do I do? If you can solve this one, promote yourself to "Neuro-Surgeon, First Class."

Wendell G. Bartlett
N. Anson, ME

A. You guessed right! Basic is doing a garbage collection to recover fragmented string space. Basic does this any time it does not have enough contiguous string space to save the new string that you are trying to store. Garbage collection is the process by which Basic shifts all available free bytes to one end of the string space. This involves determining the location and length of every string that is in string space and shifting it to a new location. This new location will be higher in memory than the old location. If Basic still cannot find enough string space, an OS error occurs.

The next logical question would be, "How does string space get fragmented?" Any time a string is stored Basic uses only enough bytes of memory to hold the characters in the string plus 1 byte for the string length. If you then change the value of the variable so that it is smaller, Basic will put the string in the same location. However, the string will not fill all the bytes previously used. If the new string value is longer than its old value, Basic cannot put it in the same location because there is not enough space and it has to store the string someplace else. In both of these situations, a hole is left in string space which is not recovered until a garbage collection occurs.

I suggest two ways to minimize this. The first is to force all your strings to be a fixed length. To do this you need to fill the unused bytes with some character. A good choice for this character is the ASCII null (CHR$(0)). Remember that the null is a valid character and must be included in any

IFs comparing strings. The second technique is to POKE the strings directly into memory using the ASC function.

Q. I own a CoCo with 64K RAM, two disk drives and a DMP-200 printer. I would like to buy an 8K to 64K buffer for my printer. I do not know of any companies that produce buffers for the Color Computer that are compatible with my DMP-200. Could you give some suggestions?

Ken Bobel
Heber Springs, AR

A. Radio Shack is now marketing its PT-64 Printer Controller #26-1269, $249.95. It has parallel inputs and outputs. Its output will mate with the parallel port on your printer, but you will need a serial-to-parallel converter, such as the one made by Botek Instruments, to hook the Printer Controller to the CoCo's serial port. A much cheaper route would be to use a software print spooler, of which there are a number on the market in the $30 price range.

Q. I have a 16K standard CoCo which I would like to expand to 32K and Extended Basic. Can you tell me where to buy the chips and the approximate cost? Do instructions come with them? I have done quite a bit of kit building and putting projects together from scratch. My machine has 1.1 Basic and an E board.

Alan R. Mac Hattie
Scotia, NY

A. Adding Extended Basic only involves plugging the Extended Basic ROM into the empty socket that is provided on the board. Since you have an E board, you will need to run a few jumper wires and bend the pins on a few chips for 64K. For 32K, you need only install the 4164 dynamic RAMs and set the jumpers to the 32K position. The 4164 chips should cost about $5 each (you will need eight) for the generic type or $69.95 for the Radio Shack kit #26-3017. Radio Shack sells the Extended Color Basic kit #26-3017 (ROM and manual) for $39.95. You will probably get a lot of resistance from your Radio Shack store about not being able to purchase these items without the "required installation," but there are many independent dealers advertising in this magazine that are more than willing to accommodate you.

Q. On my 16K CoCo, when I punch in the speedup POKE 65495,0, I cannot use the "O" key. Is this a symptom of my computer or is it shared by all CoCos?

Michael Mumford
Disputanta, VA

A. I have not experienced the problem except with my disk plugged in, but I know of others who have. As to a solution, I will speak to the D board machine. Other machines, F and later, have different component numbers. Three capacitors, C85, C73, and C75 are placed right across the clock lines, E and Q to ground. On some CoCos, this pulls the clock rise and fall time so far out of spec that when the address-dependent mode (your POKE) is invoked, the clock can't get itself straight fast enough to access the PIAs.

```
10 INPUT"START";S
20 INPUT"END";E
30 INPUT"FILENAME";F$
40 F$=F$+"/BAS"
50 OPEN"O",1,F$
60 PRINT#1,"10 FOR I="S" TO "E
70 PRINT#1,"20 READ X"
80 PRINT#1,"30 POKE I,X"
90 PRINT#1,"40 NEXT I"
100 L=50
110 FOR I=S TO E STEP 8
120 L=L+10
130 A$=STR$(L)+" DATA "
140 FORJ=0 TO 6
150 A$=A$+STR$(PEEK(J+I))+","
160 NEXTJ
170 A$=A$+STR$(PEEK(J+I))
180 PRINT#1,A$
190 NEXTI
200 CLOSE#1
```

Program Listing 1. DATA-Statement Generator

One solution is to try the following: Cut capacitors C85 or C73 and C75. They can be replaced with 10–15 pF capacitors to keep the noise down if necessary. If it still doesn't work, replace the CPU with a faster 68A09 or 68B09. Finally try replacing the PIAs with 68A21 or 68B21s. I solved the disk problem on my machine by cutting capacitor C85 and leaving the other capacitors intact.

For more information, see 80 Applications, by Dennis Kitsz, *80 Micro*, August 1982, p. 352 and "Clean Screen for CoCo," by Howard Bassen, *HOT CoCo*, July 1983, p. 102. *Do not attempt this if you are not skilled.* If it doesn't work, and it may not on all machines, you may need to restore the parts to their original configuration to get your machine to work at all.

Q. I noticed that many machine-language programs are published with POKE routines for those without assemblers. How are these generated?

Jose Perez
New York, NY

A. One way is to read the machine-language code from the left side of an Assembly listing and transcribe this information into DATA statements. I used to do this, but I found it tedious and error prone, so I developed Program Listing 1 to do it. To use my program, you need to position it so that it does not occupy the same place in memory as the machine-language code. A Basic program such as this can be positioned high in memory with a PCLEAR8 or low with a PCLEAR0 (POKE25,6:NEW or POKE25,14: POKE&HE00,0:NEW with disk). Once your PCLEARing is finished, run this program and it will generate an ASCII Basic program on disk. If you change the #1s in the PRINT statements to #−1s and delete line 40, it will work with a tape system.

Q. My friends told me that there is a POKE to disable the break on the CoCo. Please tell me the POKE to turn it off and the POKE to turn it back on again.

John J. Wender
Panama City, FL

A. Turning it back on is easier than turning off. To turn it off, run Program Listing 2. To turn it back on, type the following statement:

POKE&H19B,&H82: POKE&H19C,&HB9

```
10 FOR I=&HF8 TO &HFE
20 READI$: POKE I,VAL("&H"+I$):
NEXTI
30 DATA 32,62,1C,AF,7E,AD,A5
40 POKE&H19A,&H7E: POKE&H19B,&H0
: POKE&H19C,&HF8
50 PRINT"BREAK DISABLED"
```

Program Listing 2. Routine to Reenable the Break Key

Q. I have a standard Basic CoCo, and the SQR function does not seem to work. I always get zero for an answer. What's wrong?

Joe Crosby
San Diego, CA

A. The SQR function is supplied with Extended Basic. The computer is accepting your SQR(2) as the second subscript of the subscripted variable SQR. You can still calculate square roots using Program Listing 3.

```
10 INPUT X
20 IF X<0 THEN PRINT "<0 ERROR":
END
30 IF X=0 THEN Y=0: GOTO 80
40 Y=X/2
50 Y0=Y
60 Y=(Y*Y+X)/(2*Y)
70 IF ABS(Y-Y0)/Y0>.0001 THEN 50
80 PRINTY
90 PRINTY*Y
```

Program Listing 3. Routine to Find a Square Root Using Color Basic

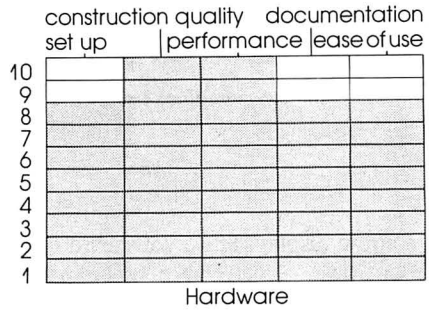
Q. I need an 80-column display. Have you heard of a hardware mod for the CoCo?

Chris Beard
Lawrence, MA

A. An 80-column card for the CoCo called Word-Pak is marketed by PBJ Inc., P.O. Box 813, N. Bergen, NJ 07047. It is available fully assembled and tested, or you can purchase the bare board and supply your own components. They also market patches so that Flex and OS-9 can use their display. In order to use their board, you will need a monitor, and if using it with disk, you will also need a Y-cable or the Radio Shack Multi-Pak interface or equivalent. ■

See List of Advertisers on page 99

REVIEWS



CONTENTS

edited by Mark E. Reynolds

Word-Pak
PBJ Inc.
P.O. Box 813
North Bergen, NJ 07047
$139.95
$19.95 patch to Frank Hogg's Flex (disk)
$19.95 patch to OS-9 (disk)

by Terry Kepner

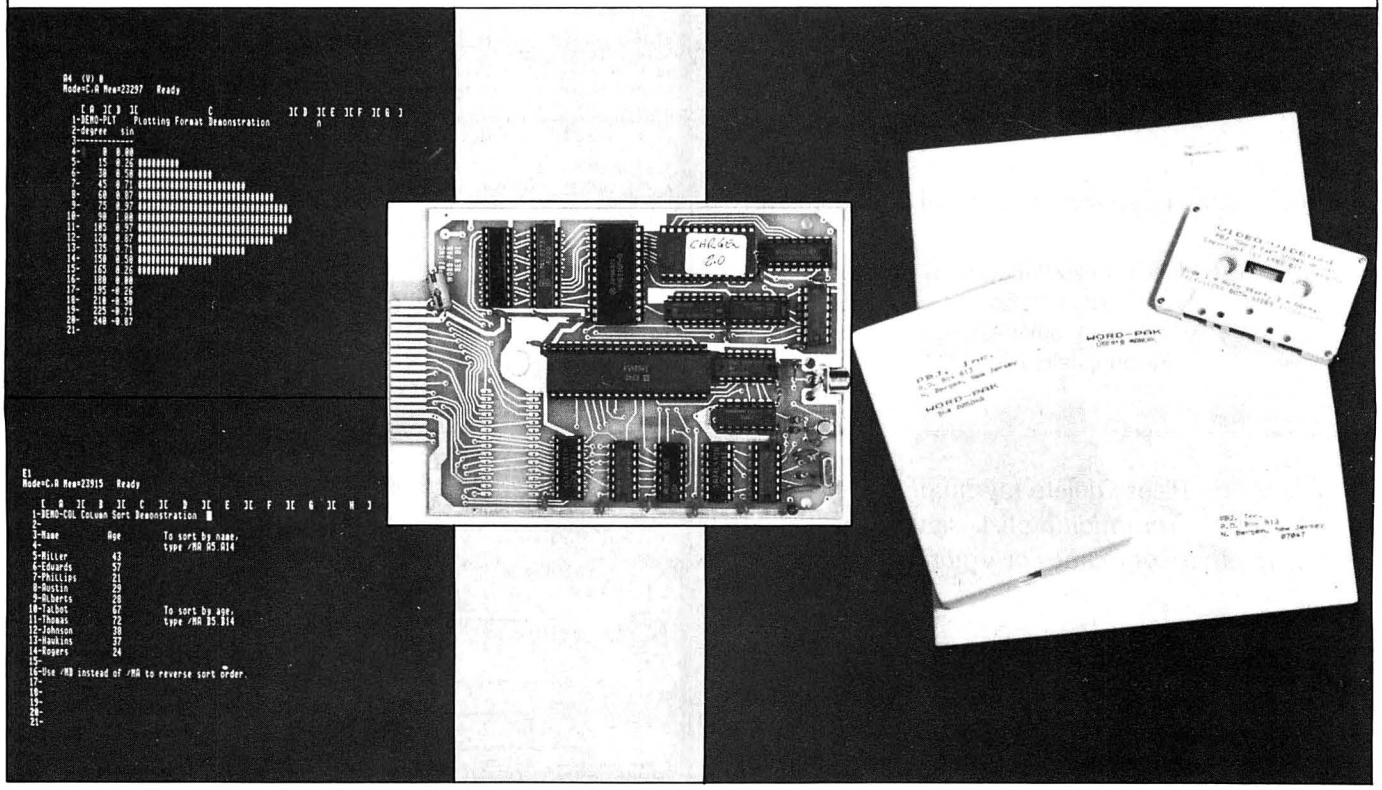
Word-Pak is a hardware device with special driver software that hooks your Color Computer to a standard monitor, giving it an 80-character-by-25-line display, true upper- and lowercase ASCII characters, and many graphics characters. You get a 390 percent increase in video display—from 512 characters to 2,000.

To use Word-Pak, plug it into the CoCo's cartridge port and load the self-executing software. It sends all Color Computer commands that display text on the video (PRINT, TAB, INPUT, etc.) to your monitor, but it sends graphics commands (*LINE*, CIRCLE, DRAW, etc.) to the Color Computer video output jack.

The Word-Pak driver program uses only 1,536 bytes of available memory, and the cartridge has its own 2K video RAM buffer. The program tests your computer for 16K or 32K and places itself in the appropriate RAM slot. If you have a 64K machine, you can load that version of the program, which will copy the Basic ROMs over to the 64K upper bank and then relocate itself to $F000 through $FEFF.

This is compatible with both disk and tape systems, so you shouldn't notice any difference in operation. Both versions change the reset-button vector so that pressing the button won't turn off the Word-Pak program. I tested the program with 16K, 32K (piggy-backed RAM), and 64K computers—all Model I versions—with no difficulties.

Now, if Word-Pak only gave you 80-column-by-25-line display, it would



still be worth the money, but it does much more: It supports the full ASCII character set, a set of graphics characters in an 8-by-8 matrix (like the Model I/III graphics characters), a programmable screen format (you change the screen width from 80 characters per line to any of seven other screen widths, including the standard 32 characters), a programmable cursor, the ability to use clear as a control key, and 18 control codes to manipulate text.

One of these control codes claims to offer double-size characters, but this is a bit of a disappointment, because it doesn't change the size of the letter at all. It merely inserts a space between each character printed.

Word-Pak supports the Color Basic PRINT@ command, so you can use normal Color Computer programs by specifying a 32-characters-per-line screen width. You can use other widths, too, but the PRINT@s won't line up properly unless they were written for the larger screen size.

The ability to change the number of characters per line is useful for designing software to use different computer-display parameters. Word-Pak lets you select line lengths of 32, 36, 40, 42, 51, 64, 72, and 80 characters, but the screen isn't remapped. Changing from 80-character to 32-character lines only affects the lines typed after you issue the command, leaving the previously typed 80-character lines intact (until they scroll off the screen).

Word-Pak also offers a welcome key-repeat function. Holding down any key for more than a few seconds causes it to repeat at a rate you can set with the escape (CHR$(27)) command, or you can disable it entirely. You can hold down more than one key at a time and have them repeated in sequence, like this: emememem.

Word-Pak gives you control of the size format and blink rate of the cursor, or you can eliminate it entirely. You can also design your own custom character sets and replace the EPROM in Word-Pak with one better suited to your needs. The documentation includes complete instructions on designing these characters, and a program to help you. You also get technical instructions on the direct programming of Word-Pak's LSI CRT controller chip.

These technical instructions include a parts schematic, parts layout, and the source code for Word-Pak's driver program (which was written and copyrighted by Cer-Comp).

"This is compatible with both disk and tape systems..."

You can get software patches with Word-Pak that let you use the driver program in OS-9, Frank Hogg's Flex, and CP/M operating systems for the Color Computer, nicely covering the disk market.

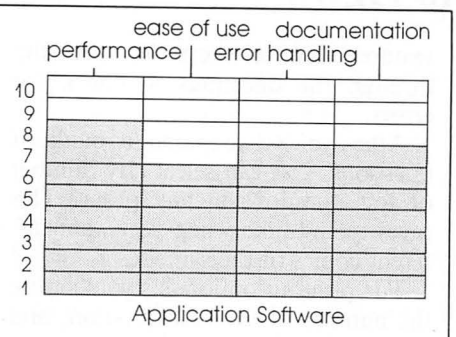
Unfortunately, Word-Pak has its faults. First, if you want to use it with a disk system (or other cartridge-based device), you must have either a multiport expansion device, or a Y expansion cable for simultaneously connecting both cartridges.

Second, the Word-Pak cartridge shuts off normal video. This means you must type without seeing your entry on the screen when trying to load the driver software from disk or tape. This will be a problem if you aren't a careful typist.

Third, most monitors can't display graphics unless the graphics have been designed as characters. And Word-Pak doesn't support the color monitors that support line graphics; therefore, Word-Pak doesn't permit color displays, because it treats the color monitor as a B/W monitor. But the program does contain a set of 32 graphics characters, to give you some graphics abilities on your monitor. These include the common variations of Word-Pak's 2-by-2 block in the 8-by-8 matrix, and all the edge variations of a 4-by-4 block in the same 8-by-8 matrix.

One interesting side effect of this graphics incompatibility is that you can use a standard monitor and a color TV at the same time. Word-Pak's driver software automatically routes all the text to your monitor, while it sends all the graphics commands to your color TV. So you *can* have your cake and eat it too.

I was surprised to find that the Word-Pak cartridge uses tin-lead connectors instead of the higher quality silver ones. Otherwise it is a well-constructed piece of equipment. Word-Pak frees you from the constraints of the tiny 32-character-by-16-line display without putting a big dent in your budget. ■



Test-Aid
Infotools
111 Country Club Lane
Oxford, OH 45056
32K
Extended Color Basic
$21 cassette
$23 disk

by James Wood

One of the little bothersome aspects of teaching involves creating and maintaining good tests. It might not be too hard to come up with a pop quiz, but what about a midterm or final exam? And the task is especially difficult if you happen to be a department chairman or some other poor soul who has to produce parts of a school- or district-wide competency or achievement test.

And what's the life expectancy of even the best test? Inevitably the answers circulate through the student grapevine.

Infotools' enhanced version of Test-Aid doesn't solve all these problems, but it does help you create, store, and print multiple-choice tests. It lets you write, edit, and select questions; store or load them to tape or disk; and create a heading for the test. The version I saw required four answer choices per question, but you can request a five-answer version.

You can write questions of up to 250 characters, with a maximum of 50 questions per file. When you type in questions and answers, the program will split any text that is longer than the number of spaces on the screen. The material will start on one line and end on the next—there is no word-wrap. And the created test comes on paper—students can't take the exam on the computer.

If you make a mistake typing in a question or answer as you design the test, you can correct it with the edit

feature. Control keys let you flip through the questions to check for errors.

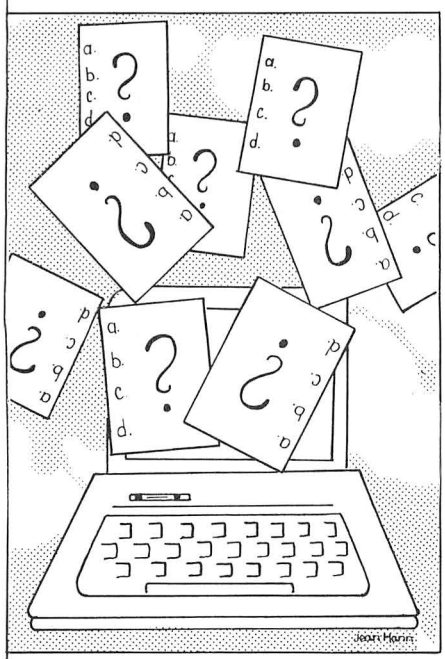
After you've prepared a bank of questions, you can select any number of test questions in any order. This saves typing time when you want different copies of the same test.

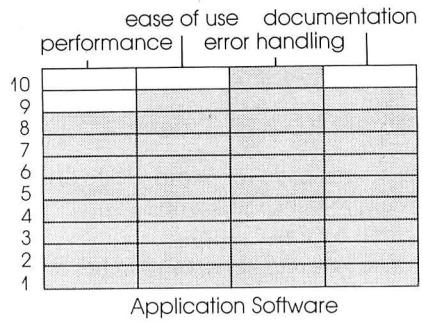
The printout option lets you choose the number of the first question, and you can choose and renumber questions from different files.

When trying Test-Aid, I couldn't return from the display-and-select-questions option to the main menu, although the instructions say that the clear key performs this function. But you can handle errors by typing GOTO 40, so after displaying questions, I press break and then type GOTO40 to return to the menu.

The program prints each answer, no matter how short, on a different line. This sometimes wastes paper, and it limits you to a maximum of 10 one-line questions on an 8-by-11-inch sheet of computer paper.

Once you have developed a store of questions, however, Test-Aid lets you manage, select, and switch them around to easily allow several tests from the same material. Creating good multiple-choice tests is no easy task, and this program is helpful, especially once you've developed the questions. Its most serious flaw, though, is that it won't grade the papers for you. (Just kidding.)■





Order Entry System
Mark Data Products
24001 Alicia Parkway No. 207
Mission Viejo, CA 92691
32K, one disk drive
80-column printer
$99.95

by Steve Brown

Mark Data's Order Entry System represents the second in what looks like a continuing series of professional, business-oriented software. I reviewed their first offering, the Business Accounting System, in last month's *HOT CoCo*.

The Order Entry System is made up of an interactive set of programs that allow a business owner to enter orders received from customers, generate invoices for the products ordered, and track sales on a periodic basis. This easily used, menu-driven system leads you logically through each task. The software is comprised of a series of interlinked programs, rather than one long, complex program.

Screen Environment And Error Handling

Included on the disk is Mark Data's Superscreen (reviewed in *HOT CoCo*, January 1984, p. 40). Superscreen is a machine-language utility that replaces the standard 32-character-by-16-line screen display with a 51-character-by-24-line screen, and a full upper- and lowercase character set.

Superscreen also adds an ON ERROR GOTO statement that traps errors and avoids crashes and loss of data. When the program encounters an error, it branches to a designated routine that halts the program but retains all data and displays the type of error. This error avoidance is especially valuable in a business program in which money, not hobby time, is at stake.

Ease of Use

The Order Entry System's menu leads you through a series of programs that let you use all or just some of the system's features. The programs included create, update, and maintain data files and printout invoices and sales reports. One utility even includes a line feed for those printers that don't issue one with each carriage return.

Order Entry

You can easily enter orders, but you must do a bit of work to initialize the system before you enter them. First, you must run the NEWFILES program, which creates the three-system data files. Then you must run SYSTMFIX, a program that lets you access the SYSTEM file, containing system-support information such as company name, current date, current invoice number, and total open and closed orders.

The other programs in the Order Entry System use and sometimes modify the data in the SYSTEM.DAT file. For example, the date in the file is changed whenever you use the system, and the invoice counter advances as you enter each transaction. Mark Data has also made provisions for you to modify or correct the data in the system-data file.

You must also enter all the products you sell and the unit price, so you can automatically generate invoice prices and dollar sales reports. Fortunately, you only need to perform this setup once, and then the system updates itself whenever you run it.

The Order Entry menu lets you enter a new order, review both closed and open orders, or close an open order. The menu screen also tells you the total number of closed and open orders in the data file.

You enter an order by typing data into a formatted order form. As you proceed, you're given the option of changing sales conditions, discounts, and tax codes, which adds a nice flexibility to order entry. Finally, the finished order goes to the printer to create a complete invoice.

You can print your own invoice format, but Mark Data has set up the system to use a standard, commercially available form that you can purchase at any business-forms store. You can then add your own letterhead. The program also lets you prepare sales

Continued on p. 101

HOT CoCo

Index to Advertisers

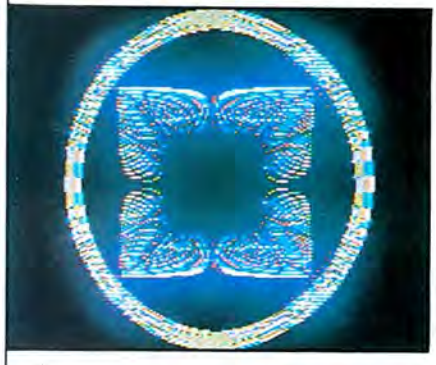
*This advertiser prefers to be contacted directly.

For further information from our advertisers, please use the Reader Service card.

COMING NEXT MONTH

While tanned landscape painters, brush and paintbox in hand, are capturing summer countryside, the CoCo artist will find plenty of indoor inspiration in the August Graphics Issue of *HOT CoCo*.

You can study graphics techniques in simplified form, and create some beautiful screens with "Mini Graphics Fun." This feature is ideal for beginners and interesting to experienced computer artists.



Once you have mastered artistic technique, you'll find helpful advice on how to store and recall graphics screens from disk in "Let the Gypsy Lady Show You How." A savings of 105K per disk gives you no small graphics advantage.

August is storm season, and if hurricanes interest you, whether you're safe in New Hampshire or battening down the hatches on the Gulf Coast, our "Hurricane Tracker" program and article is going to fascinate you. Find your storm and watch her progress with the help of your CoCo.

The hacker: calm, methodical, technical. You depend on *HOT CoCo* for features, and our next issue won't disappoint you. "ROM Hacker—Part III" continues its technical exploration.

HOT CoCo readers might not be tanned by next month, but you'll be entertained and informed.

Subroutines To Go

Help yourself to *Machine-Language Subroutines for the Color Computer*. It's a library of useful ready-to-use machine-language routines. Each subroutine is on the cassette that is included in the package. Many of the routines can be used in BASIC as well as machine-language programs. You'll find ROM subroutines, which are located in the interpreter ROMs of the Color Computer, and RAM subroutines, a collection of routines written by David McLeod.

What's inside

Each chapter contains a specific category of routine, giving a six-letter filename for each routine, a brief description of the routine and what it does, entry and exit requirements and a program listing or sample call. You'll learn about Color BASIC 1.1, Extended BASIC 1.0, Disk BASIC 1.0, numeric conversions, data processing routines, keyboard input routines, text output using high-resolution graphics, tape/disk input/output and multiple precision routines.

Easy to use

Machine-Language Subroutines for the Color Computer is packaged for ease of use while you are working at the computer. Each major section is marked with an index tab. Appendixes and an index of subroutines are included. If you are a machine-language programmer, you'll find *Machine-Language Subroutines for the Color Computer* a valuable addition to your program library.

Machine-Language Subroutines for the Color Computer
David D. McLeod ISBN 0-88006-070-0 CC7404 8½ × 11
$29.95 cassette included

To order

To order, call toll-free *1-800-258-5473* for credit card orders. Or mail your order with check or money order or complete credit card information to: *Wayne Green Books, Retail Sales, Peterborough, NH 03458*. Please include $3.00 per book for shipping and handling. WGBooks are also available at your local bookstore. Dealer inquiries invited. Available May 1984. Please allow 4-6 weeks for delivery. Orders are payable in U.S. dollars only.

331

☐ **Yes . . . Send me** _____ copies of *Machine-Language Subroutines for the Color Computer* (CC7404) at $29.95 each, plus $3.00 per book for shipping and handling.

☐ payment enclosed ☐ MasterCard ☐ Visa ☐ Am. Ex.

Card # MC bank #

Expiration date Signature

Name

Address

City State Zip

Wayne Green Books, Peterborough, NH 03458

Continued from p. 98

reports for any period desired.

Sales Reporting

The program generates a two-part report. Part 1 of the sales report lists record number, invoice number, sale date, type of sale, net dollar sale, tax, shipping charges, C.O.D. charges, and invoice total for each invoice. It also gives dollar totals for all invoices generated in the reporting period.

Part 2 yields bank-deposit information such as total cash sales, checks received, and bank-card sales. This simplifies daily deposit preparation and checks it against filled orders. Next, the program generates an item-number list of all items sold and includes the quantities of each item and extended dollar sales. Finally, it tallies total sales, total discounts granted, and total net sales.

Documentation

Instructions for using the system are excellent. The comprehensive manual is divided into three parts: One explains each portion of the system, one tells how to learn and customize the system to your needs, and one explains how to use the system regularly for order processing. You also get clear directions for customizing the system to fit your business and equipment configuration.

Performance

The system performs smoothly and the error-trapping routine takes away much of the worry over costly mistakes. You can prepare professional-looking invoices and reports, and handle and change files quickly.

The Order Entry System has a number of other nice features. For example, when you enter orders, you can apply different discounts or tax rates to that single order.

Mark Data has obviously given much thought to making the programs fit a wide range of printer baud rates, line feeds, form feeds, and so on, each of which you can change.

Viewed strictly as a simple order entry system—such as a mail-order operation might use—the Mark Data system does a creditable job. If you usually enter and fill all orders from stock, and you usually receive cash, check, or credit-card payments rather than leaving an open account, the system is complete enough.

"Instructions for using the system are excellent."

However, if orders usually contain a number of items, some of which might be backordered, or if you put many transactions on open-account payment terms, you'll find some shortcomings in the system. For example, if a customer phones to inquire about an invoice, you must go find the printout; you can't look up a specific invoice number in the data files.

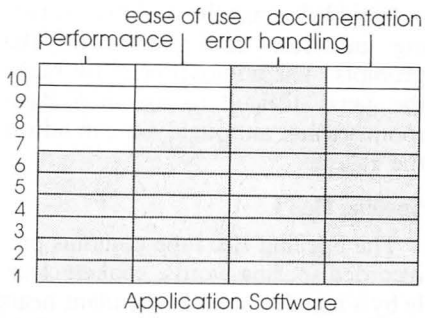
Perhaps more limiting, there are no provisions for accessing transactions by customer account number or by customer name. Thus, you can't go to the computer and determine what open orders exist for a particular customer. You access transaction records on the disk by finding the record number, not the invoice number. No doubt this is a result of the program's particular file-handling and sorting routines.

Neither can you invoice for partial shipments, since the invoice is generated at the time of order entry, before you physically fill the order. Of course, you could return the partially filled invoice to the order-entry station and create two new invoices—one for shipped items and one for backordered items, but this method offers a greater chance for error.

The Order Entry System also doesn't provide a way to produce a duplicate invoice for a transaction already entered.

Many of these features are included in larger and much, much more expensive software packages for small mainframe computers. Now Mark Data appears to be building a library of business programs, each member addressing a particular need. If that is the case, then the Order Entry System could serve as a link in that chain.

Taken on its own merits, the Order Entry System is an easy way to generate invoices and track sales. It doesn't have the bells and whistles of larger systems, but it requires much less expense and computing power. If your business needs an order-entry system to help keep records and avoid the clutter of orders and notes scratched on matchbook covers, and you'd like to add professionalism to your enterprise, the Mark Data system is well worth its nominal price. ■



TCE Language Package
TCE Programs Inc.
Box 2477
Gaithersburg, MD 20879
Spell Bomber
16K, Extended Color Basic
$18.95 cassette
$22.95 disk
Spelling Bee I
16K, Extended Color Basic
$16.95 cassette
Synonym Express
16K
$14.95 cassette

by John Steiner

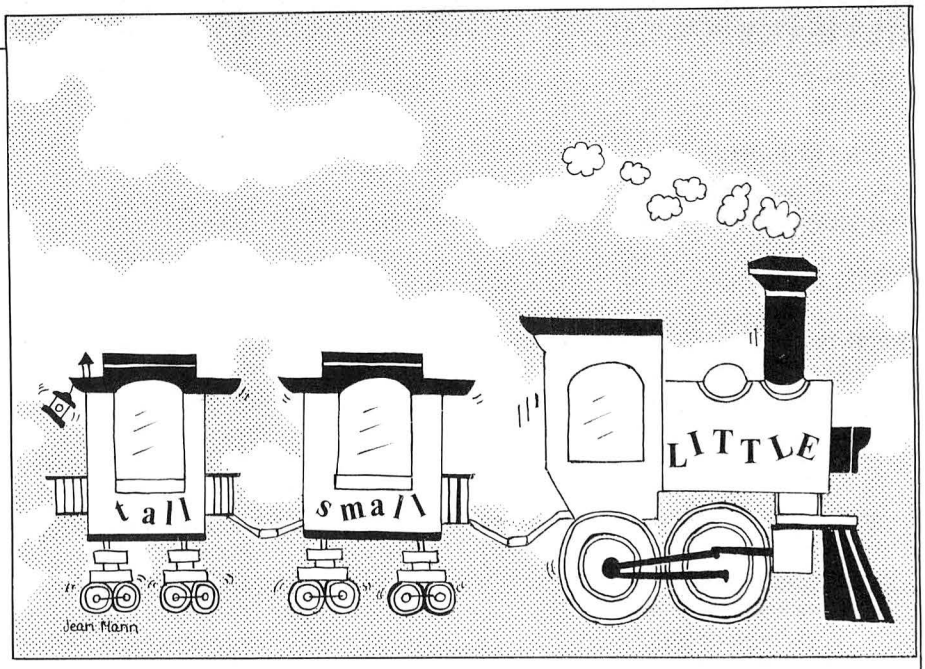
The three programs in TCE's Language Package for first and second grades are good examples of inexpensive, Basic-language educational software. They come individually boxed, and each uses the Color Computer's graphics and audio features effectively.

Spell Bomber

Spell Bomber requires Extended Color Basic and uses medium-resolution, four-color graphics to display a battleship and an enemy bomber. The game is a variation of the old favorite, Hangman. The student gets only eight tries to guess the correct letters to fill the blanks on the screen. With each wrong guess, the bomber moves closer to the student's ship.

If you fill the blanks within eight tries, the ship calls General Quarters, and its cannons destroy the bomber. If you don't guess the word, though, the bomber destroys the ship. The screen displays the score after each word.

Three skill levels make the game more challenging for older students. Technically, it is not a spelling teacher as much as it is a guessing game. Since the higher difficulty levels will definitely be a challenge with only eight guesses, you could use Spell Bomber with the higher elementary grades.



I couldn't crash the program by typing incorrect information at the prompts. The audio effects will make the game difficult to use in a classroom setting, although you can adjust the volume.

Spelling Bee I

The Spelling Bee tape contains prerecorded spelling words, spoken clearly by a female voice. The student hears a spelling word, which he must then type into the correct number of blanks that appear on the screen.

If he spells it correctly, a screen display praises him using his name. If he makes a mistake, the correct spelling is displayed briefly at the bottom of the screen. The program responds to a second error by displaying the word a bit longer. After three errors, it displays the word until the student types it correctly, and then a whale eats the word.

Although there are many words on both sides of the tape, the word list is fixed, so you can't change it unless you know something about Basic programming. And the whale takes much too long to eat the word once the student gets it right. Spelling Bee would be much better if you could add your own words to the tape.

The program is well done and has nice graphics. Input-error trapping keeps the lesson on track. Because the whale eats the word so slowly, however, and because you see it over and over, the program might easily lose a first or second grader's attention.

Synonym Express

Synonym Express is my favorite of the three programs. It uses a simple concept, has interesting graphics, and runs in Color Basic. As it begins, a train with a word hung on the side of the engine chugs onto the screen. It pulls two cars behind, each carrying a word. The student must choose the word that is synonymous with the word on the engine.

Each time the student is incorrect, the screen displays the rule for synonyms. Since there are only two choices, you only get one try for a correct answer.

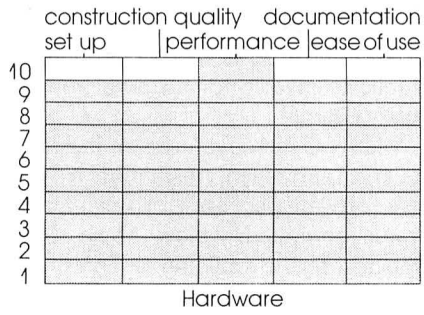
The animated character-string graphics are striking and show what a good programmer can do with Color Basic, but the program can become tedious after a while, because it presents each word in the same way. This doesn't seem to be a lesson you could use frequently.

The words are in DATA statements within the program, so anyone who understands the DATA statement could easily add new choices. The only sound is that of the Choo Choo, which shouldn't be too disruptive in a classroom setting, if you keep the volume down.

These programs use the sound educational principles of immediate reinforcement and repetitive drill, with graphics and audio enhancements. More variety and, in some cases, faster action would be an improvement. The elementary student's attention span is short, and things have to happen quickly to keep him interested.

These programs would also be more useful if you could modify them and add your own word lists more easily. The classroom teacher should be able to adapt educational software to reinforce particular units.

The Language Package does, however, provide some inexpensive training and practice for young students. ■



Serial/Parallel Interface
PBH Computer Products Inc.
P.O. Box 55868
Houston, TX 77055
$89.95

by John Steiner

Does the printer you want to buy have a parallel interface, making it useless for the CoCo? Or, have you had it with plugging and unplugging your modem and printer from the single serial I/O jack? The PBH interface offers a handy solution to both problems.

The interface consists of a small box from which come four cables. On one cable is a male serial I/O port that plugs into your CoCo, on one is a female four-pin connector that plugs into a modem or other RS-232 device, and on the third is a standard 34-pin parallel connector that you can attach to any Centronics-type printer. The fourth cable has a small AC adapter that plugs into the wall and powers the unit.

There is a printer/modem/off switch on the side of the box, and one on top that selects printer baud rates from 300–9,600.

Before you connect the interface to the computer, you must determine whether you have the Color Basic 1.0 or 1.1 ROM version. On Color Basic machines, you can read the sign-on message when you turn on your computer. However, the sign-on message

of Extended Color Basic machines only gives you the version of Extended Color Basic (or Disk Basic). You must type in EXEC 41175 to get the Color Basic sign-on message.

If you have the 1.0 ROM version, you can open the PBH unit and reposition a pair of jumpers. The manual gives a careful explanation of this procedure.

If you have Color Basic 1.1 or 1.2, the unit works properly as shipped. Set the baud-rate selector to 600, and LLISTs and PRINT #-2s operate nearly all Centronics-compatible printers. If you're using a modem, just put the printer modem/off switch in the modem position; you don't need to unplug the printer.

The 600-baud printer port is slow, and one of PBH's advantages is that you can operate it at 9,600 baud, which noticeably improves printing speed. There is only one catch: You must tell the computer what baud rate to use. Do this by using a POKE command, or check with many software packages that tell you how to change the printer baud rate in their print routines. The memory locations that you must modify for different baud rates are as follows:

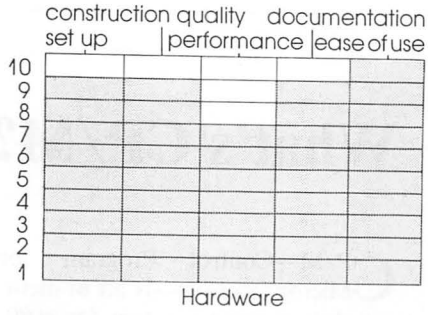
Baud Rate	Data
300	180
600	87
1,200	41
2,400	18
4,800	7
9,600	1

POKE 150, DATA

To use the table, look up the desired baud rate, set the PBH baud-select switch accordingly, and enter POKE 150, DATA (DATA is the number next to the desired baud-rate column). For example, to select 9,600 baud, enter POKE 150,1. After execution, the new baud rate stays in the computer until you change it, or shut the computer off.

The PBH documentation is a three-page booklet containing set-up information for all aspects of the system and a useful troubleshooting chart that helps pinpoint installation problems.

PBH is new to CoCo accessories, but there's no inexperience reflected in the interface. The workmanship and packaging are both excellent, and the unit has performed well on my 1.1 Color Basic machine. ■



CoCo Coupler
Wayne Technology
P.O. Box 5196
Anaheim, CA 92804-1196
64K, one disk drive
$250

by Terry Kepner

The CoCo Coupler cartridge lets your 64K disk-based Color Computer run the more than 10,000 programs designed to operate with the CP/M 2.2 standard. It converts your 6809E CPU machine into a Z-80 CPU computer operating at 4 MHz instead of .89 MHz. You simply plug the cartridge into the cartridge port and plug your Radio Shack disk drive into the CoCo Coupler. A single keypress selects either CP/M capability or standard CoCo Disk Basic.

"If you're a skilled 8080 or Z-80 Assembly-language programmer, or want to be, the CoCo Coupler lets you create and run 8080 code programs."

When you are running CP/M, you have access to the disk operating system (DOS); you can't access Basic, as you can with the normal Color Computer DOS. This is an important difference: If you want to program in Basic, you'll have to spend several hundred dollars on Microsoft's MBASIC, or find a public-domain Basic somewhere. And if you want the CoCo's color capabilities, you'll need Microsoft's GWBASIC.

If you're a skilled 8080 or Z-80 Assembly-language programmer, or want to be, the CoCo Coupler lets you create and run 8080 code programs. A public-domain CP/M assembler that generates Z-80 code already exists. Consider that you can buy a disk-based Color Computer system with CoCo Coupler for under $1,000, whereas an equivalent 8080 or Z-80 system can cost from $1,500–$2,000, or more.

Inside the CoCo Coupler

Designing CP/M for the Color Computer couldn't have been easy, since the typical CP/M display is 80 characters by 24 lines, and CP/M doesn't run on the 6809 CPU system. Wayne Technology added a special utility, VIDEO80, that treats the Color Computer's standard 32-by-16-line display as a window on the CP/M 80-by-24-line display. You use the arrow and clear keys to maneuver the window around the larger display.

To further insure capability with standard 80-column displays, the CoCo Coupler configures the TV display circuitry to act as a Lear-Siegler ADM-3 display terminal, the most popular on the market. The Coupler also supports control codes that let you reposition the cursor and direct the cursor-addressing routines so you can specify the line and column where you want to start printing. The cartridge uses the ASCII character set, beginning with 32, for logic.

Wayne Technology has also altered the keyboard interpretation to disable the break key, use clear as a control key, and give the four arrow keys the following three functions: they move the cursor; when used with the clear key in the VIDEO 80 utility, they move the display window around the 80-by-24-line screen; and when shifted, they select the left and right brackets, generate the ESC code, and home the cursor.

The CoCo Coupler will also support 40- and 80-track disk drives (in addition to 35-track) and 18 124-byte sectors (the Omikron Model I CP/M format) per track (single or double density).

Wayne Technology's CP/M is licensed from Digital Research Inc., who also designed the Basic Disk Op-

erating System (BDOS) and the Console Command Processor (CCP). Wayne Technology designed its own Basic Input/Output System (BIOS), which integrates the Color Computer's-Input/Output operation with the normal expected operation of these functions in CP/M.

The BIOS, BDOS, and CCP are the main sections of CP/M and reside in memory at all times. The rest of memory, 40K on the Color Computer, is called the Transient Program Area (TPA), and you use it for your application programs.

CP/M is usually distributed with a package of TPA programs that make the system much more useful. The Wayne Technology package includes the standard CP/M features and all the standard CP/M programs: ASM, DDT, DUMP, ED, LOAD, PIP, STAT, SUBMIT, SYSGEN, and XSUB. You also get: SETUP.COM, FORMAT.COM, VIDEO80.COM, WORDPAK.COM, CONFIG.COM and TRANSFER.COM.

The CP/M Utilities

SETUP.COM is a simple program that lets you select lowercase-character generation from the keyboard, lowercase display on the video, and the RS-232-port baud rate. I was happy to find that CP/M accepts the lowercase command inputs as proper commands, instead of giving you a syntax error like TRSDOS and Microsoft Basic do.

FORMAT.COM formats 35-, 40-, or 80-track, single-density disks in 18 128-byte sectors per track. Naturally, you can't format a disk with more tracks than your disk drive can support.

VIDEO80 converts the 32-by-16-line Color Computer display into a window on the normal CP/M 80-by-24-line display. You definitely need this utility, since the screen is otherwise difficult to read. With large directories, some of the entries will scroll off the screen before you can read them. Also, some programs won't work with a screen size other than 80-by-24-line.

WORDPAK.COM is a specialized utility for Wordpak, an 80-column display cartridge from PBJ Inc. (See the Wordpak review elsewhere in the Review section.) The Wordpak cartridge requires that you use a standard

What's CP/M?

CP/M—Control Program for Microprocessors—is an in-memory disk operating system for 8080, Z-80, 8086, and 68000 CPU-based microcomputers. Actually, it was the first comprehensive disk operating system (DOS) for 8080 and Z-80 computers. And now it's available for the Color Computer.

But just what is CP/M, and what's its significance for the Color Computer? That requires a little history.

CP/M History

CP/M came from a need for a disk operating system for microcomputers, back when they were still only the toys of computer hobbyists and professionals. In those days, Bill Gates was still in college, Atari was an unknown, the Apple I was just being developed, and Radio Shack hadn't even thought of the Model I.

A few pioneers in the industry had managed to build computer systems that included not only a CPU, but could also run a video display and let you input data and instructions from a keyboard (instead of using rows of switches). Someone had even managed to connect disk drives to the machine and transfer data between the two.

One programmer took such a system and wrote a program that supervised the operation of the computer, monitor, keyboard, printer, and disk drives. This original DOS design was for an 8080 Altair computer.

The program controlled the computer so effectively that other people began asking for copies of it. Finally, Digital Research Inc. began selling the DOS. The fledgling computer industry fell in love with this new, convenient, available DOS, and it rapidly spread across the country.

It spread so rapidly because it was designed to be device-independent. That is, the CP/M system used set addresses for sending and receiving information to and from the various peripherals. A new computer manufacturer merely had to honor those addresses to be compatible with CP/M and CP/M programs.

Of course, different computers had different peripherals, some with wider displays, or better keyboards, or whatever, requiring different driver machine-code instructions, but CP/M doesn't care what the actual driver code looks like, only that the driver respond to the proper CP/M data address.

This means that a program written on one CP/M computer, as long as it honors the CP/M input/output (I/O) calls, will run on almost any other CP/M computer. So if your machine is from a new company, or even if you designed and built it yourself, you still have many CP/M programs available for it. With a non-CP/M computer, you have to wait for programmers to buy the machine and then start writing programs for it.

The CP/M system is so popular then, because, as the oldest DOS, it has the most, and some of the best, programs written for it. (WordStar is the best selling word processor in the world, and VisiCalc was originally developed for CP/M.) Somewhere, someone has probably already written the business or application program you want. In many cases quality CP/M programs are public domain, free for you to use and modify to your heart's content.

The CP/M System

CP/M is divided into four separate sections, the Basic input/output system (BIOS), the Basic disk operating system (BDOS), the command control processor (CCP), and the transient program area (TPA). Digital Research writes the BDOS and CCP for each computer system a manufacturer designs, while the computer manufacturer writes the BIOS to put their computer's hardware into CP/M parameters.

The BIOS contains the actual machine-code necessary to send characters from CP/M to the display screen, send characters from the keyboard to CP/M, connect the RS-232 port to CP/M, connect the parallel port to CP/M, and perform other

miscellaneous hardware-dependent tasks. The BDOS and CCP provide the software environment for the CP/M programs. While the BIOS varies from manufacturer to manufacturer, the BDOS and CCP will always appear to be the same to any CP/M program in the computer.

Finally, the TPA is the memory area available for CP/M programs. In most computers, this is about 40K (out of 64K) of RAM, although some versions of CP/M are limited to 28K (for 48K computers), and 12K (for 32K computers).

CP/M comes on disk with a set of built-in commands, TPA system utilities, and special utilities from the computer manufacturer to use with CP/M on your particular machine.

CP/M loads completely into your computer's memory. Then you can remove the system disk and replace it with a data disk. If you need certain utilities, you can put them on the data disk also, but you don't have to have a system disk in drive 0 (or drive A, as CP/M calls it) at all times.

You can now run programs, examine the directory, erase files, rename files, save files, or list files on the current drive. The term "current drive" is used because CP/M works only with the drive you last selected. That is, when you run your program, you start with drive A. Until you specify otherwise, all disk actions are restricted to the disk in this drive. If you want drive B, then you must specify so in the CP/M command. You can switch the current drive to any in the computer system, which then restricts all disk I/O to that drive, unless otherwise specified.

The disadvantage to this approach is that an OPEN command in Basic will search only the current drive, and not all the drives on the system. For a complete search, you have to specify each drive in turn, an awkward process.

CP/M has only the following six built-in commands (built-in means that all the code for those commands resides in memory at all times):

- DIR (get a directory),
- ERA (erase the selected file),
- REN (rename a file),
- SAVE (save the file in memory to disk),
- TYPE (list the selected file), and
- USER (for multi-user systems—not always available). Running a program simply involves typing its name.

The following are standard CP/M system utilities:

- ASM (an 8080 Assembly-language assembler),
- DDT (Dynamic Debugging Tool—a machine-language monitor),
- DUMP (list the specified file in hexadecimal format),
- ED (a line-oriented text editor also used for making files for ASM assembling),
- LOAD (takes ASM-created files and makes them into executable code and adds the suffix .COM),
- PIP (the peripheral-interchange program, a very versatile copy program),
- STAT (gives statistics on the current disk or specified program),
- SUBMIT (executes a file of CP/M commands, like a DO file on the Model III and 4),
- SYSGEN (copies CP/M to a new disk), and
- XSUB (allows programs to accept file input as keyboard input—usually used with the SUBMIT command).

Notice that Basic isn't considered a CP/M utility program. If you want Basic for your computer, you must either purchase it from a commercial source or find a public-domain version.

Many times the computer manufacturer will supply other utility programs that customize CP/M operation to their computer, including, in some cases, Basic.

If you're interested in CP/M, several books give good descriptions of the various commands and how to use them. The two best are *The Osborne CP/M User Guide* by Thom Hogan, from Osborne/McGraw Hill Books, and Rodnay Zaks' *The CP/M Handbook, with MP/M*, from Sybex. And the Nanos Systems Corporation (P.O. Box 24344, Speedway, IN 46224) has a 20-panel CP/M system reference card for only $5.95—*Terry Kepner*■

"Having the entire 80-by-24-line display on the screen is quite an improvement over windowing."

black-and-white monitor instead of a TV, but gives you a clean, sharp 80-column-by-24-line display in exchange.

Using Wordpak with Wayne Technology's CP/M requires a multipak-expansion device, or a Y cable that lets you attach two cartridges to the CP/M hardware cartridge in your Color Computer. Unlike standard Wordpak operation, you have normal TV display until you run WORDPAK.COM which switches display from the TV to the monitor.

If you use Wordpak and WORDPAK.COM, you don't need VIDEO80. Having the entire 80-by-24-line display on the screen is quite an improvement over windowing.

CONFIG.COM configures the disk drives on your Color Computer, so CP/M will automatically know that your Drive C is an 80-track drive, for example. It also lets you select the printer baud rate and printer-busy line (some printer's hold this line high when they're busy, so this feature can prevent character loss during printing).

TRANSFER.COM lets you transfer programs and data through the RS-232 port. The data format is the same as the public domain program, MODEM (also known as XMODEM). This format is 8-bit words, no parity, one-stop bit, and no data compression.

Finally, Wayne Technology CP/M includes a feature that lets you perform single- and double-drive backups and copies.

The Problem with RFI

In spite of its otherwise good design, with copper edge connectors and a clean parts layout, the Coupler cartridge has one fatal flaw: It isn't properly grounded against Radio Frequency Interference (RFI).

The Radio Shack disk systems have small metal tabs on either side of the cartridge connector. These tabs fit in-

to spring clips on the Color Computer's main Printed Circuit Board (PCB). These tabs properly ground the cartridge to the computer, eliminating a serious RFI problem on your TV when you're using the disk drives. Otherwise, RFI would cover the TV screen with snow, making it difficult to read.

The CP/M cartridge also has these two tabs, but it doesn't have clips on the other end into which the disk cartridge plugs. Thus, while the CP/M cartridge is properly grounded to the computer, the disk cartridge isn't grounded and produces snow.

If you use a multipak-expansion device, you make the problem even worse, since you must plug the CP/M cartridge into the computer and the expansion device into the CP/M cartridge. With the disk cartridge in the expansion box, the CP/M video display is almost impossible to read. Even normal TRSDOS operation leaves the TV screen unreadable.

This flaw in grounding design renders the TV display annoying to use in straight operation (CP/M cartridge and disk cartridge only), and impossible to use with a multipak-expansion box.

"...the Wayne Technology device is an inexpensive way to tap into the huge CP/M program market."

The only way to use the CP/M cartridge, multipak, and disk cartridge at the same time is to purchase the Wordpak cartridge and use a standard video monitor instead of a TV. A standard monitor doesn't have the circuitry required to receive broadcast signals, and, therefore, doesn't pick up the RFI produced by the disk cartridge. The display is, as a result, clear and sharp.

This lack of grounding also voids the Color Computer's design to conform to the FCC regulations about RFI pollution.

Another criticism (minor or major, according to your workspace) is that the total system almost doubles the width of the computer, since you plug the disk cartridge into the CP/M cartridge, adding a 10-inch extension to the right side of the computer. With a multipak, the space requirements are even greater.

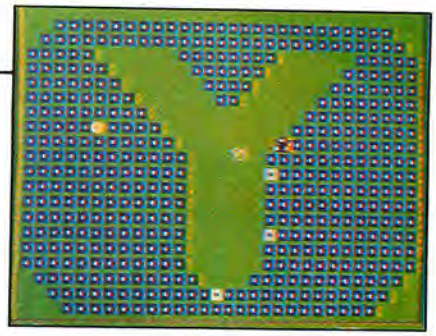
All in all, if you use the CP/M cartridge with only a disk cartridge, or if you use the Wordpak cartridge and a monitor with your expansion box, the Wayne Technology device is an inexpensive way to tap into the huge CP/M program market.

If the CoCo Coupler and the programs that come with it don't expand your Color Computer horizons enough, you might be happy to hear that Wayne Technology is almost ready to release a new utility that lets you read from and write to 12 other disk formats. In fact, you should be able to modify it to adapt to any format. ■

Gameware

This month's games take you from the outer fringes of the Milky Way, across the conveyor belts of a candy factory, to the most inaccessible and terror-ridden corridors of an underground labyrinth. It's quite a trip.

Dungeons of Daggorath



Beam Rider

Radio Shack

Dungeons of Daggorath, a new, 3-D, real-time adventure from Radio Shack (1400 One Tandy Center, Fort Worth, TX 76102, cat. no. 26-3093, $29.95 16K ROM pack) has captured my fascination as few other games have.

To begin, you, as the neophyte adventurer armed with a torch and wooden sword (would you take anything less?) are transported to the dungeons beneath the mountains of Daggorath to meet and destroy an evil wizard who has enslaved the land. But the wizard has plenty of unfriendly underlings who will greet you along your way.

You have an adventurer's-eye view of the corridors and doorways in the dungeon while the bottom of the screen displays your heartbeat (don't let it beat too fast—or stop) and tells you what you hold in each hand. You can enter most commands and object words as single letters.

As you move through the dungeon, different creatures will attack. Each creature makes a characteristic sound as it roams the halls, and you usually hear them before you see them. However, because the game is in real time, when a monster enters your space to attack, you must strike and move quickly to avoid his blow and defeat him. The one- and two-letter commands come in handy.

But be careful—if you enter commands too quickly, and consequently move too fast, your heartbeat will increase to the point at which you may pass out. On the other hand, if you move too slowly, you'll be an easy target for a creature's blow.

You gain strength from each opponent you kill, and some of them will be carrying things—better weapons, magical objects, and extra torches—that make your quest easier.

Dungeons of Daggorath intrigues me. The long, torchlit passageways are realistic enough to keep me wondering what I'm going to meet next. And it's a real challenge to defeat the stronger creatures. I have often been frustrated, though, by the fact that there can be a one- or two-second delay between the time you type in a command and the time it appears on the screen. Therefore, I often don't know if I've made a typo, or if keys have failed to register, until it's too late. In combat situations, each command is critical.

Spectral Associates

Beam Rider is an original idea in fast arcade action from Spectral Associates (3420 South 90th St., Tacoma, WA 98409, $24.95 16K tape or disk).

The screen displays a pattern of closely packed blocks and some open space. Using the joystick, you try to guide a "beamer" over as many blocks as possible, while avoiding the chasers and spinner. The spinner bounces randomly around the screen, but the chasers attempt to get in your way.

Your beamer moves rather quickly from block to block. If it moves across open territory, it automatically shoots a beam across to the first block in its path and doubles its speed along the beam. It goes very slowly when it moves into open space that isn't directly towards a block.

This sounds simple enough, but this game requires not only quick reflexes, but also careful planning and strategy. The joystick control is good enough to allow precise movement, but, as always, arrow-key control would be a welcome option in these games in which you can only move up, down, right, and left.

Beam Rider is a good, fast, strategy game. And perhaps most impressive of all, it's an original idea in a sea of video clones.

Color Panic ($24.95, 32K tape or disk) is the most frustrating video game I've ever played. That doesn't mean it's bad—certain minds will love the careful planning it requires, but I just don't have such a mind.

The idea behind it is that a lone human is stranded on a distant planet that happens to be inhabited by giant, deadly mutants. The planet looks like a four-story building without walls, and ladders connect from two to three floors, but never all four at once. Therefore, getting from the ground to the fourth floor means you'll have to use more than one ladder.

And of course, the mutants pursue the poor castaway relentlessly. His weapon? A shovel. So you thought a wooden sword was bad.

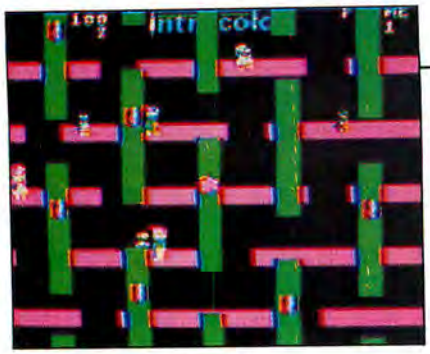
And now the frustration comes in—at least for me. Your human begins on the bottom floor, with mutants on the other three. He must climb to an upper floor and stand in one place long enough to dig a trap for the mutants, avoiding their lethal touch all the while.

Hopefully, an unsuspecting mutant will fall into the hole, and the castaway has about five seconds to get back to the hole and fill it in before the creature climbs out as a more deadly foe.

The human is definitely at a disadvantage. He has no powerful phaser pistol, nor any special running or jumping ability. You've got to plan your moves carefully in order to outwit a superior adversary. To make matters worse, you can't move your human up or down a ladder unless you have him in precisely the right spot—the one place I can never find.

Well, the graphics in Color Panic are good, and it certainly is a challenge. Like I said, some people will love it.

Color Panic



Candy Co.

Moon Shuttle

Intracolor

Candy Co. (Intracolor, P.O. Box 1035, East Lansing, MI 48823, $34.95 32K tape or disk) is a lighthearted idea in which you must move Candy Dan (in the single-player mode) and Q. P. Doll (two-player mode) across the conveyor belts of a candy factory, picking up candy while avoiding the bad guys, the cherry bombs, and the sticky stuff.

But as hard as it might be to take this scenario seriously, the game presents some very real logistical problems.

All action on the screen takes place on five vertical and five horizontal moving conveyor belts. Belts parallel to each other move in opposite directions, and each belt is broken into sections, so you can't run along it from one side of the screen to the other. Certain objects on the belts are obstacles or hazards.

At any given moment, a belt Candy Dan is standing on could be touching one, two, three, or none of the belts perpendicular to it. If you want him to pick up an object on the belt beside him, you've got a real problem trying to catch up to it. And if you do manage to get near, the object might hop onto another belt.

Candy Co. is an entertaining game. It offers great graphics, lighthearted fun, and a serious planning challenge.

Datasoft

I first looked at **Moon Shuttle** (Datasoft, 9421 Winnetka Ave., Chatsworth, CA 91311, $29.95 32K tape and disk) as a sideways Space Invaders, or a souped-up version of Datasoft's Pooyan. In it you use the joystick or arrow keys (yay!) to control a space ship that appears at the left of the screen. You must move it from screen to screen in which you must either blast your way through an asteroid belt or destroy waves of quick-moving aliens.

Getting through the asteroid belts isn't too tough, even at the higher levels, but the dodging, firing aliens are another story. Even the first wave isn't exactly easy, and it's not long before you'll find yourself in some of the fastest arcade action you're likely to see.

Moon Shuttle has Datasoft's fine graphics and nice packaging (the box includes both tape and disk versions of the game). It doesn't demand the planning skills that you'll need for the other games this month, but it does require the quickest reflexes and concentration.—M.E.R. ■

Tips

Do you have a hot tip on a game, or need one? Share your discoveries and frustrations here.

Lancer Pause

Here's an undocumented pause feature in Spectral Associates' Lancer: Press break to pause and shift/clear to continue.

Matt Lawson
Bartlesville, OK

Pyramid Puzzles

How can I get through the maze in Radio Shack's Pyramid, and how can I get past the snake?

Kent Jakway
Garrett, IN

Pyramid Pointers

Here are some clues that will help you find that last treasure in Radio Shack's Pyramid.

You must give the M...his two treasures or you will never find the chest. Once you enter the maze, you're only eight turns from the treasure chest. Six moves in the proper order, and you are at a pit, and the maze continues on that level.

Use "down" on the way out to find the treasure, and use everything you can carry to the maze, to leave a bread-crumb trail of sorts. That will help you map the maze, and you'll be surprised at what you find.

Oh! What's the opposite of S/E?

Kenneth Dey
Kansas City, MO

Here You Go, Ray

Here are some tips for Ray Gallantry to use in Radio Shack's Madness and the Minotaur.

To get out of the maze with "up" and "down," keep bearing north and up. East or west just varies the exit point. After three moves in any given direction, you tend to return to the starting point.

Once you get to the maze area that doesn't let you move up or down (where the "mist obscures the wall"), exit is fairly easy. After moving west through the string of rooms that have only east/west exits, go S, S, S, E, S, E, N, N, E, E, N, W, and then jump the pit. See Table 1 for a list of spells.

Now, can someone tell us how to get the last two treasures? We've often reached 220 points, and I'm sure the answer has something to do with the pack-rat. Anyone have a suggestion?

Ruth E. Chaffin
Mentor, OH

And Again

To get out of the maze in Madness and the Minotaur, locate the area in which a layer of mist obscures the east wall and jump over the mist. Then go north and west. That should bring you to a room with a small pit in the corner. Jump the pit and you will be in the great forest.

For help on the spells, Ray Gallantry will have to be more specific.

Paul Riddle
Linthicum, MD

Table 1. Spells for Madness and the Minotaur

Nymph	Mushroom	Flute	Food
	Truthring	Okkan	Pendant
Troglodyte	Ax	Spellbook	Dagger
	Scepter	Crom	Shield
	Mitra		Nergal
Sprite	Food	Skull	Dagger
	Nergal	Powerring	Akhirom

Table 1 continued

Table 1 continued

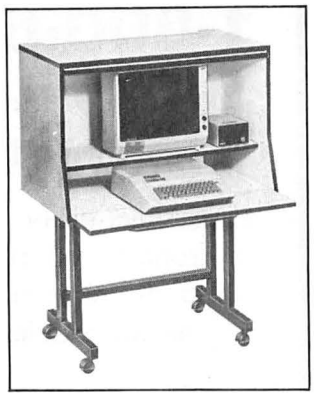
Scorpion	Skull Crom	Flute Nergal	Talisman Powerring		
Satyr	Sword Nergal	Spellbook Vetar	Spellbook Crom	Mace Lightring Mitra	Mace Powerring Mitra
Minotaur	Mace Vial Scepter Mitra	Sword Shield Powerring	Ax Nergal		
Powerring	Belrog	Shield Mace Skull Mitra	Shield Scepter Parchment Okkan	Shield Skull Mitra	Shield Sword Akhirom
Truthring	Lightring Ax Okkan	Lightring Flute Akhirom	Lightring Basket Crom		
Lightring	Powerring Talisman Crom	Powerring Rope Nergal	Powerring Ax Okkan		
Spellbook	Skull Flute	Pendant Crom	Mushroom Goblet Belrog	Powerring Nergal	
Vial	Talisman Mitra	Mushroom Vetar	Parchment Mitra		
Skull	Talisman Pendant	Scepter	Okkan		
Sword	Rope	Mitra	Parchment		
Shield	Dagger	Sword	Mace		
Vetar	Food/Mushroom				
Mitra	Parchment				
Okkan	Talisman				
Akhirom	Rope				
Nergal	Scepter				
Belrog	Vial				
Crom	Pendant				
Ishtar	Spellbook				
Maze	From area with no up/down after area with only east & west, go S, S, S, E, S, E, N, N, E, E, N, W, then jump pit.				

✓ See List of Advertisers on page 99

PRODUCT NEWS

edited by Cynthia Smith

Information used in the Product News section is supplied through manufacturers' press releases. *HOT CoCo* has not tested or reviewed these products and cannot guarantee any manufacturer's claim.



The Big Max computer workstation from Hubbard Scientific Company.

Computer kitsch for coffee break.

Good-bye Hunt-and-Peck!

Good typing skills can save you hours of tedious work entering programs. ETT (Electronic Typing Teacher) by Cherrysoft is designed to help you type with confidence.

ETT's video keyboard lets you practice with all keys labeled, all keys blank, or only the home keys labeled. The visual cues guide you while you learn to type without watching your fingers. ETT keeps score and times you so you can see improvement.

With the sentences provided by ETT, learning to type can be fun. They include every letter in the alphabet and you get a fresh set every time you run ETT. You can also create your own practice sets to test yourself.

ETT shows your accuracy, response time, and words-per-minute, and comes with a 10-page student manual/study guide and cassette tape. It requires 16K Extended Basic and retails for $21.95. Contact CoCo Warehouse, 500 North Dobson, Westland, MI 48185. 313-722-7957.

Reader Service ✓ 553

Roots

Explore your roots with this genealogy program from Autumn Color Software. Ancestors uses direct-access disk files to create, modify, and display up to 500 genealogical records. The index displays or prints names and assigned reference numbers, and you can display each complete record on screen or print it as a typical family group sheet. You can display a three-generation pedigree chart on the monitor, and page forward or backward from any designated starting point.

Each record holds 22 fields of data including record and reference numbers, name, number of children, dates and places of birth, marriage, death and burial as well as occupation, military, religion, and residence information. Also, data on other spouses and reference to parents is included in each record.

Ancestors is available on disk for $39.95 and includes a fully documented manual. It requires 32K and one disk drive and can be ordered from Autumn Color Software, 4132 Lay St., Des Moines, IA 50317.

Reader Service ✓ 560

Big Max

Hubbard Scientific Company's new line of lockable workstations offers a choice of computer-compatible styles and prices to house and protect hardware and software, and provides an efficient working environment.

Big Max and Little Max models secure keyboards, monitors, disk drives, and software under lock and key to prevent theft, accidental or malicious damage, and unauthorized use.

Built of 3/4-inch wood with wear-resistant work surfaces, and tubular steel frame construction, the units are available with or without locking casters.

They meet recommended standards for correct keyboard and monitor heights, and provide plenty of extra space for paperwork and software. Four-outlet power strip and power-cord organizer trays are built into the back of the units.

The line also includes single and double level work tables for flexible use in housing self-contained or multiple-component systems. For more information contact Jack Langlois at T. Coleman Associates, 1151 Rush St., Gary, IN 46403. 312-272-7810.

Reader Service ✓ 551

Mr. Dig

Computerware has a new high-resolution arcade game for the Color Computer.

In Mr. Dig you have to help the little wizard harvest his cherry crop by guiding him through the orchard and away from the bad hunters. If they get too close, squash them with falling fruit or bounce your magic orb their way. If you capture the extra treats you get extra points and chompers that chase Mr. Dig. Don't forget the bad lettermen. Eliminate all five of them and you get an extra Mr. Dig and go on to a new screen.

Mr. Dig costs $27.95 on cassette and $30.95 on disk plus $2 shipping, and requires 32K and a joystick. Contact Computerware, Box 668, 4403 Manchester Ave., Suite 103, Encinitas, CA 92024. 619-436-3512.

Reader Service ✓ 555

Sweet Gum's Magic Mug

Lighten up your coffee break with a little computer kitsch. The "Magic Computer Input/Output" mug is one of the computer compatible gift items offered by Sweet Gum Inc.

Pouring hot liquid into this mug makes one message disappear and another appear on the monitor. Currently available with messages such as "Programmer On/Off Duty," "Computer Expert On/Off Duty," and "Computer Nut On/Off Duty," the 10-ounce mug retails for $8.95, plus $2 shipping.

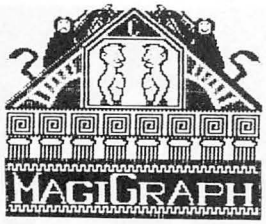
The free Sweet Gum catalog features a binary clock, computer related tee-shirts, samples of antique core memory, a microchip jigsaw puzzle, and other novelty items.

Order the current catalog or the mug from Sweet Gum, 15490 N.W. 7th Ave., Miami, FL 33169. 305-687-9338.

Reader Service ✓ 552

Speed Math

Speed Math from West Bay Company is a math learning game designed to give practice and encourage speed and comprehension



NEW GOOD STUFF FOR EVERY COLOR COMPUTER

Turn your Color Computer into a graphic design center with the ease of a keystroke! **MagiGraph** makes it simple to create highly detailed figures up to and including an entire high-resolution screen. Designed for those with some experience in Basic and Assembly Language programming, **MagiGraph** includes lots of special features:

- **A full set of logical and pixel manipulation functions** simplifies the development of complex figures.
- **An editor** lets you zoom in and work on every detail of your design. Toggle between the "macro" and "micro" screens for perspective on your creations.
- **Nine animation buffers** allow you to preview each sequence to ensure continuity and smooth flow.
- **Versatile I/O routines** store a graphic screen on cassette or floppy disk; recall it later for use by another program or revise it with **MagiGraph**.

If you're looking for the finest graphic development utility available for your Color Computer, THIS IS IT. Maximize your machine's potential, while you push your imagination to the limit — with **MagiGraph**!

By Kevin Dooley. Cassette **$34.95** (16K required); Disk **$39.95** (32K Extended Color BASIC required); Amdisk cartridge **$44.95.**

CSPOOL
Color Computer Print Spooler

Stop Waiting Around for the Printer! **CSPOOL** allows you to use your printer and computer concurrently, takes only 26 bytes of Color Basic's memory, and gives you 32K of print buffer. It's like having two computers in one! By intercepting characters sent to the printer and storing them in the upper 32K of RAM, **CSPOOL** allows you to run other programs while your printer is doing its job. **CSPOOL** is FREE with the purchase of a 64K RAM UPGRADE KIT from The Micro Works, or it may be purchased separately on cassette or diskette for **$19.95**. Requires 64K; not for FLEX or OS9.

64K MEMORY UPGRADE KIT: For Rev. levels E, ET, NC, TDP-100s, and Color Computer II. Eight prime 64K RAM chips, instructions, and CSPOOL: **$64.95.**

SYSTEMS SOFTWARE

MACRO-80C: DISK-BASED EDITOR, ASSEMBLER AND MONITOR—With all the features the serious programmer wants, this package includes a powerful 2-pass macro assembler with conditional assembly, local labels, include files and cross referenced symbol tables. MACRO-80C supports the complete Motorola 6809 instruction set in standard source format. Incorporating all the features of our Rompack-based assembler (SDS-80C), MACRO-80C contains many more useful instructions and pseudo-ops which aid the programmer and add power and flexibility. The screen-oriented editor is designed for efficient and easy editing of assembly language programs. MACRO-80C allows global changes and moving/copying blocks of text. You can edit lines of assembly source which exceed 32 characters. DCBUG is a machine language monitor which allows examining and altering of memory, setting break points, etc.

Editor, assembler and monitor--along with sample programs—come on one Radio Shack compatible disk. Extensive documentation included. By Andy Phelps. **$99.95**

SDS-80C: SOFTWARE DEVELOPMENT SYSTEM—Our famous editor, assembler and monitor in Rompack. Like MACRO-80C, it allows the user to write, assemble and debug assembly language programs with no reloading, object patching or other hassles. Supports full 6809 instruction set. Complete manual included. **$89.95**

MICROTEXT: COMMUNICATIONS VIA YOUR MODEM! Now you **can** use your printer with your modem! Your computer can be an intelligent printing terminal. Talk to timeshare services or to other personal computers; print simultaneously through a second printer port; and re-display text stored in memory. Download text to Basic programs; dump to a cassette tape, or printer, or both. Microtext can be used with any printer or no printer at all. It features user-configurable duplex/parity for special applications, and can send any ASCII character. You'll find many uses for this general purpose module! ROMPACK includes additional serial port for printer. **$59.95**

MICRO WORKS COLOR FORTH

- Faster to program in than Basic
- Easier to learn than Assembly Language
- Executes in less time than Basic

The MICRO WORKS COLOR FORTH is a Rompack containing everything you need to run Forth on your Color Computer. COLOR FORTH consists of the standard Forth Interest Group (FIG) implementation of the language plus most of FORTH-79. It has a super screen editor with split screen display. Mass storage is on cassette. COLOR FORTH also contains a decompiler and other aids for learning the inner workings of this fascinating language. It will run on 4K, 16K, and 32K computers. And COLOR FORTH contains 10K of ROM, leaving **your** RAM for **your** programs! There are simple words to effectively use the Hi-Res Color Computer graphics, joysticks, and sound.

Includes a 112-page manual with a glossary of the system-specific words, a full standard FIG glossary and complete source listing.

MICRO WORKS COLOR FORTH . . . THE BEST! From the leader in FORTH, Talbot Microsystems. **$109.95**

MACHINE LANGUAGE

MONITOR TAPE: A cassette tape which allows you to directly access memory, I/O and registers with a formatted hex display. Great for machine language programming, debugging and learning. It can also send/receive RS232 at up to 9600 baud, including host system download/upload. 19 commands in all. Relocatable and reentrant. **CBUG TAPE: $29.95**

MONITOR ROM: The same program as above, supplied in 2716 EPROM. This allows you to use the entire RAM space. And you don't need to reload the monitor each time you use it. The EPROM plugs into the Extended Basic ROM Socket or the Romless Pack I. **CBUG ROM: $39.95**

SOURCE GENERATOR: This package is a disassembler which runs on the Color Computer and generates your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. A 16K system is required for the use of this cassette. **80C Disassembler: $49.95**

HARDWARE

PARALLEL PRINTER INTERFACE—Serial to parallel converter allows use of all standard parallel printers. PI80C plugs into the serial output port, leaving your Rompack slot free. You supply the printer cable. **PI80C: $59.95**

SUPER-PRO KEYBOARD—$69.95 (For computers manufactured after Oct. 1982, add $4.95)

ROMLESS PACKS for your custom EPROMS — call or write for information.

BOOKS

6809 ASSEMBLY LANGUAGE PROGRAMMING, by Lance Leventhal, $18.95

TRS-80 COLOR COMPUTER GRAPHICS, by Don Inman, **$14.95**

ASSEMBLY LANGUAGE GRAPHICS FOR THE TRS-80 COLOR COMPUTER, by Don Inman, **$14.95**

STARTING FORTH, by L. Brodie, **$17.95**

GAMES

ZAXXON—The real thing. Excellent. What more can we say? Cassette requires 32K. **$39.95**

STAR BLASTER—Blast your way through an asteroid field in this action-packed Hi-Res graphics game. Available in ROMPACK; requires 16K. **$39.95**

PAC ATTACK—Try your hand at this challenging game by Computerware, with fantastic graphics, sound and action! Cassette requires 16K. **$24.95**

HAYWIRE—Have fun zapping robots with this Hi-Res game by Mark Data Products. Cassette requires 16K. **$24.95**

ADVENTURE—*Black Sanctum* and *Calixto Island* by Mark Data Products. Each cassette requires 16K. **$19.95** each.

CAVE HUNTER—Experience vivid colors, bizarre sounds and eerie creatures as you wind your way through a cave maze in search of gold treasures. This exciting Hi-Res game by Mark Data Products requires 16K for cassette version. **$24.95**

THE MICRO WORKS

P.O. Box 1110-D
Del Mar, CA 92014
(619) 942-2400 ✓196

California Residents add 6% Tax

Master Charge/Visa and COD Accepted

✓ See List of Advertisers on page 99

in addition, subtraction, multiplication, and division.

You have menu selection choices of number of problems, desired difficulty level, and timer control. Problems are presented for solution, and graded with errors identified for the user, all in a game atmosphere.

Speed Math is available from West Bay Company, Route 1, Box 666, White Stone, VA 22578, $8 (postpaid).

Reader Service ✓ 556

Advanced Adventure

Sorcerer of Claymorgue Castle is now available on tape for Color Computer adventure buffs.

Designed for the experienced adventure player, it is considered to be the most advanced of Adventure International's series. Following a Medieval magic theme, the player becomes Beanwick, apprentice to Solon the Master Wizard. He moves through a given scenario using traditional verb/noun commands to search for the 13 Stars of Power.

Priced at $19.95 (tape) it is available from Scott Adams Inc., Box 3435, 155 Sabal Palm Drive, Longwood, FL 32750-3435. 305-862-6717.

Reader Service ✓ 563

Beam On

The Beamer is a device designed to interface two Color Computers. It is not a modem, but uses a pair of wires to transfer programs and files directly from one computer to another. (For transfers you need two Beamers.)

You can also connect other personal computers in this way providing the cassette record and playback format is the analog type.

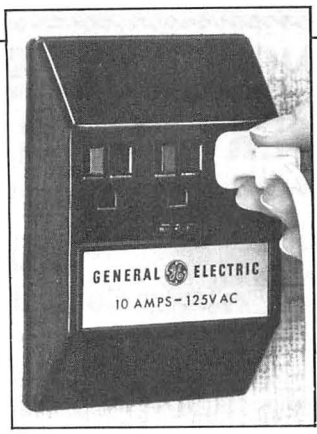
No additional software is necessary, and The Beamer uses normal CLOAD and CSAVE commands. Note that although this device is wired to match standard telephone connection, direct connection to the telephone network might be in violation of local, state, or federal laws. Owner assumes full responsibility for proper installation and operation.

Cost is $59.95 for one and $99 for two. Order from H.L. Johnson Services, 1637 Jessica Way, San Jose, CA 95121, or Silicon Rainbow Products, 1111 W. El Camino Real, Suite 109, Sunnyvale, CA 94087.

Reader Service ✓ 564

New From GE

General Electric is now offering a low-priced voltage spike detector with built-in noise filtering capabilities. Designed to protect computers, audio and video devices, and sensitive solid-state equipment from sharp spikes in electrical voltage, the GESP-753 lists for $48.75.



General Electric Surge Suppressor and Noise Filter.

It plugs into the upper outlet of any wall grounding receptacle, and a stabilizing pin fits into the grounding contact of the unused outlet to hold the unit in place. A built-in light indicates that protection circuits are working. The power cord of protected equipment plugs into one of the three grounded outlets.

For more information and product literature on the new GE GESP-753 Voltage Surge Suppressor and Noise Filter, contact the Microwave Products Department, General Electric Company, 316 East 9th St., Owensboro, KY 42301. 502-685-6200.

Reader Service ✓ 565

Solid Oak: CoCo Elegance

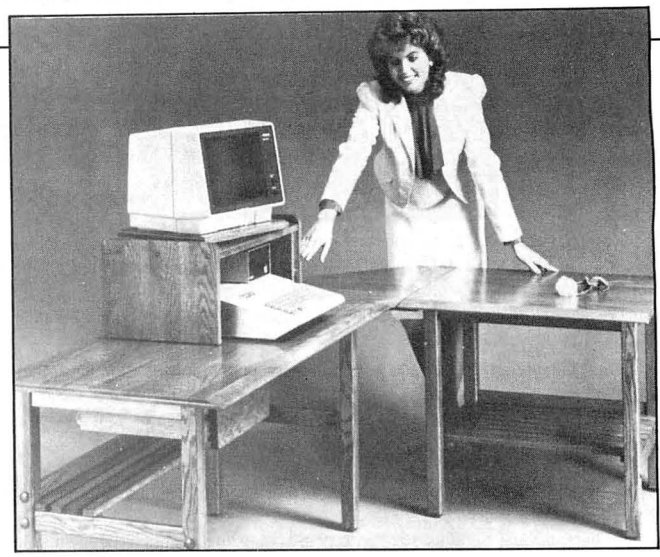
White's Computer Supplies has introduced a line of oak furniture made for the computer user. Computer desks are available in three designs that blend with any decor. The Executive is a solid oak, raised panel design. The Open Air is an open design, and the Contempo has a solid oak top with oak veneer side panels.

White's also offers a printer/plotter stand, work stations, and video swivels. Items come in honey, cinnamon, and cocoa finishes. For free literature contact White's Computer Supplies, 3348 Niles Road, St. Joseph, MI 49085. 616-429-2545.

Reader Service ✓ 558

Mul-T-Screen

Mul-T-Screen from Incentive Software prints in any size and mode using any color for a total of 10 sizes. With it you can mix up to five sizes and four colors with any colored background, on one screen. It prints vertically, runs all programs, and recognizes the clear button, CLS command, PRINT-TAB, PRINT@, and gives a character capacity from 8-by-4 to 42-by-24, while enabling you to mix and match sizes.



Solid Oak Computer Station.

Mul-T-Screen is all machine language, and includes two sample Basic programs. It requires a minimum of 16K and sells for $29.95 (tape) in Canada, $24.95 (tape) in the U.S., $32.95 (disk) in Canada, and $27.95 (disk) in the U.S. Ontario residents add seven percent sales tax. Outside Canada and the U.S., add $2.

Contact Incentive Software, Box 323, Station B, London, Ontario, Canada N6A 4W1.

Reader Service ✓ 566

Gimix DO

Gimix Inc. has licensed DO, an OS-9 shell procedure control language, from Lloyd I/O for distribution with their OS-9 system. DO is similar to Basic in appearance and reads the procedure file from disk using approximately 8.5K of user memory.

The main feature of DO is parameter passing and it passes all unrecognized statements to SHELL for execution. It has 26 number variables, nine string variables, and uses labels to control the flow of execution. The ON ERROR GOTO traps errors and allows the user's procedure to take the correct action.

DO is available for Color Computer OS-9 users for $69, from Lloyd I/O, 19535 N.E. Glisan, Portland, OR 97230. 503-666-1097.

Reader Service ✓ 562

Interface Breadboard Package

The Color Computer Expansion Connector Breadboard Model CC-100 makes it possible to connect external devices to the expansion connector signals of the computer. Combined with a solderless breadboard and the book, *TRS-80 Color Computer Interfacing, with Experiments* (No. 21893), it forms the CoCo-100 package providing interfacing instructions for any version of this computer. In addition, the CC-100 Experiment Component Package contains the parts needed for the experiments in the book.

With the CoCo-100, you can learn to access the signals available in the parallel expansion connector of the CoCo, and how to construct and use a peripheral interface adapter (PIA). The experiments demonstrate how to enter and retrieve binary data and how analog-to-digital and digital-to-analog conversion is performed both within the computer and using external devices.

Model CoCo-100, Interface Breadboard Package is priced at $51.25, a 10-percent reduction from the cost of the individual components, plus $2.50 shipping. Components of the CoCo-100 are available separately as follows:

- CC-100 Expansion Connector Breadboard: $34.95
- RS-100 Modular IC Breadboard Socket: $6.95
- Book No. 21893, *TRS-80 Color Computer Interfacing:* $14.95
- Model CC-100 Experimental Component Package: $59.95 plus $2.50 shipping (includes 16 ICs, a peripheral interface adapter, one D/A converter and one A/D converter, a binary switch, and assorted resistors, capacitors, and connectors).

Virginia residents add 4 percent sales tax. VISA and Master Cards accepted. For more information contact Group Technology Ltd., P.O. Box 87, Check, VA 24072. 703-651-3153.

Reader Service ✓ 559

See List of Advertisers on page 99

You've invested a lot of time and money into your computer . . . . .
It's time that investment paid off!

THE COLOR ACCOUNTANT

Introducing The Color Accountant (from The Programmer's Institute), the only complete personal financial package specifically designed for the TRS-80 Color Computer. This unique package includes:

1. **Complete Checkbook Maintenance**
2. **Chart of Accounts Maintenance**
3. **Income/Expense Accounts**
4. **Net Worth Statement**
5. **Payments/Appointments Calendar**
6. **Color Graph Design Package (graphs any files)**
7. **Check Search**
8. **Home Budget Analysis**
9. **Decision Maker**
10. **Mailing List**

After the initial setup, **THE COLOR ACCOUNTANT** requires less than an hour of data input each month.

The checkbook maintenance program is the key to the entire package. Once your checkbook is balanced, the checkbook summary file will automatically update the home budget analysis, net worth, and income/expense statements. You can then graph any file, record bills and appointments, make decisions, print a mailing list, analyze various accounts or stocks, and even calculate taxes.

All programs are menu-driven and allow add/change/delete. Each file and statement can be listed to screen or printer, and saved to cassette or diskette. **THE COLOR ACCOUNTANT** also comes with 40 pages of documentation that leads you step-by-step through the entire package. The TRS-80 COLOR Ext Basic requires 16K for this package.

$74.95 Cassette **$79.95** Diskette

Send $2 For Our New 64-Page Catalog (#11) Which Contains More Than 500 Software Products.
(Catalog is provided FREE with any order)

To Order:
Write or call Toll-Free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

COMPUTRONICS INC. ✓9
50 NORTH PASCACK ROAD
SPRING VALLEY, N.Y. 10977

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

AMERICAN EXPRESS MasterCard VISA

30-DAY MONEY BACK GUARANTEE

*** ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE ***
DELIVERY SUBJECT TO AVAILABILITY

BUSINESS PAC 100

Now Available for the TRS-80 Color Computer!

Includes Inventory Control, Payroll, Accounts Receivable, Accounts Payable, Checkbook Maintenance, and more. Comes with 128-page user's manual.

Interest apportionment by Rule of 78's • Annuity computation program • Time between dates • Day of year a particular date falls on • Interest rate on lease • Breakeven analysis • Straightline depreciation • Sum of the digits depreciation • Declining balance depreciation • Double declining balance depreciation • Cash flow vs. depreciation tables • Print checks with daily register • Checkbook maintenance program • Mortgage amortization table • Compute time needed for money to double, triple, etc. • Determine salvage value of an investment • Rate of return on investment with variable inflows • Rate of return on investment with constant inflows • Effective interest on a loan • Future value of an investment (compound interest) • Present value of a future amount • Amount of payment on a loan • Equal withdrawals from investment to leave zero over • Simple discount analysis • Equivalent & nonequivalent dated values for obligations • Present value of deferred annuities • Percent markup analysis for items • Sinking fund amortization program • Value of a bond • Depletion analysis • Black-Scholes options analysis • Expected return on stock via discounts dividends • Value of a warrant • Estimate of future earnings per share for company • Compute alpha and beta variables for stock • Portfolio selection model • Option writing computations • Value of a right • Expected value analysis • Bayesian decisions • Value of perfect information • Value of additional information • Derive utility function • Linear programming solution by simplex method • Transportation method for linear programming • Economic order quantity inventory model • Single server queueing model • Cost-volume-profit analysis • Conditional profit tables • Opportunity loss tables • Fixed quantity economic order quantity model • Cost-benefit waiting line analysis • Net cash-flow analysis for simple investment • Profitability index of a project • Weighted average cost of capital • True rate on loan with compensating balance required • True rate on discounted loan • Merger analysis computations • Financial ratios for a firm • Net present value of project • Laspeyres price index • Paasche price index • Construct seasonal quantity indices for company • Time series analysis linear trend • Time series analysis moving average trend • Future price estimation with inflation • Mailing list system • Letter writing system (links with mailing list system) • Sort lists of names • Shipping label maker • Name label maker • DOME business bookkeeping system • Compute week's total hours from timeclock info. • In-memory accounts payable system • Generates invoices on screen and print on printer • In-memory inventory control system • Computerized telephone directory • Time use analysis • Use of assignment algorithm for optimal job assignment • In-memory accounts receivable system • Compare 3 methods of repayment of loans • Compute gross pay required for given net • Compute selling price for given after-tax amount • Arbitrage computations • Sinking fund depreciation • Computerized UPS zone table • Type envelope with return address • Automobile expense analysis • Insurance policy file • In-memory payroll system • Dilution analysis • Loan amount a borrower can afford • Purchase price for rental property • Sale-leaseback analysis • Investor's rate of return on convertable bond

100 Ready-To-Run Business Programs

only $99.95

(Available on diskette only)
Also available for TRS-80 Models I, II and III, Apple, Atari, Osborne, Kaypro, all Commodore Computers and most CP/M systems.

The HJL-57 Keyboard

Compare it with the rest. Then, buy the best.

If you've been thinking about spending good money on a new keyboard for your Color Computer, why not get a good keyboard for your money?

Designed from scratch, the HJL-57 Professional Keyboard is built to unlock ALL the potential performance of your Color Computer. Now, you can do real word processing and sail through lengthy listings...with maximum speed; minimum errors.

At $79.95, the HJL-57 is reasonably priced, but you can find other CoCo keyboards for a few dollars less. So, before you buy, we suggest that you compare.

Compare Design.

The ergonomically-superior HJL-57 has sculptured, low profile keycaps; and the three-color layout is identical to the original CoCo keyboard.

Compare Construction.

The HJL-57 has a rigidized aluminum baseplate for solid, no-flex mounting. Switch contacts are rated for 100 million cycles minimum, and covered by a spill-proof membrane.

Compare Performance.

Offering more than full-travel, bounce-proof keyswitches, the HJL-57 has RFI/EMI shielding that eliminates irritating noise on displays; and four user-definable function keys (one latchable), specially-positioned to avoid inadvertent actuation.

Free Function Key Program

Your HJL-57 kit includes usage instructions and decimal codes produced by the function keys, plus a free sample program that defines the function keys as follows: F1 = Screen dump to printer. F2 = Repeat key (latching). F3 = Lower case upper case flip (if you have lower case capability). F4 = Control key; subtracts 64 from the ASCII value of any key pressed. Runs on disc or tape; extended or standard Basic.

Compare Installation.

Carefully engineered for easy installation, the HJL-57 requires no soldering, drilling or gluing. Simply plug it in and drop it right on the original CoCo mounting posts. Kit includes a new bezel for a totally finished conversion.

Compare Warranties.

The HJL-57 is built so well, it carries a full, one-year warranty. And, it is sold with an exclusive 15-day money-back guarantee.

Compare Value.

You know that a bargain is a bargain only so long as it lasts. If you shop carefully, we think you will agree...The HJL-57 is the last keyboard your CoCo will ever need. And that's **real** value.

Order Today.

Only $79.95, the HJL-57 is available for immediate shipment for either the original Color Computer (sold prior to October, 1982) or the F-version and TDP-100 (introduced in October, 1982), and the new 64K CoCo. **Now also available for CoCo 2.**

Order by Phone Anytime

716-235-8358

24 hours, 7 days a week

Ordering Information: Specify model (Original, F-version, or CoCo 2). Payment by C.O.D., check, MasterCard or Visa. Credit card customers include complete card number and expiration date. Add $2.00 for shipping ($3.50 for Canada). New York state residents add 7% sales tax.
Dealer inquiries invited. For dealer information in Eastern U.S. and Canada, call collect: 617-586-7614, Advanced Computer Services (distributor), 74 Plain Street, Brockton, MA 02401.

✓440

HJL PRODUCTS INC.
955 Buffalo Road • P.O. Box 24954
Rochester, New York 14624