Serving the CoCo Community for

The RAINBOW 12 YEARS

THE COLOR COMPUTER MONTHLY MAGAZINE

November 1992 Vol. XII No. 4 Canada $4.95 U.S. $3.95

The CoCo BOUNTY

a worldwide network . . .

SEE PAGE 12

The Assembly Line

The CoCo Goes Buggy

by William P. Nee

Sometimes when I'm shoveling snow or hauling firewood down to the basement, I remember those carefree halcyon days of summer in Florida. I remember sitting outside in the evening, sipping a soft drink and — you guessed it — swatting bugs! A little after sundown, the evening ritual would begin; out would come the swarms of insects and out would come the electric bug zappers. Great swarms of bugs would swirl, enticed by the pale light, and slowly revolve into it; each bug was further attracted by the faint flicker of a neighbor who had bumped into the electric coils.

To recapture this image, I decided to simulate it using the Color Computer. With *Bugs*, a large swarm of bugs appears on the screen. At first each bug moves in a random direction. But as one crosses the "finish line" of electricity, its demise attracts all the other bugs. As they move towards their late member, another bug goes — and they swirl again. Slowly the swarm gets smaller until one lone bug dances his way across the screen to his ultimate fate. Of course it takes a machine-language program to simulate all the motion and to rapidly compute the changes in direction for the swarm.

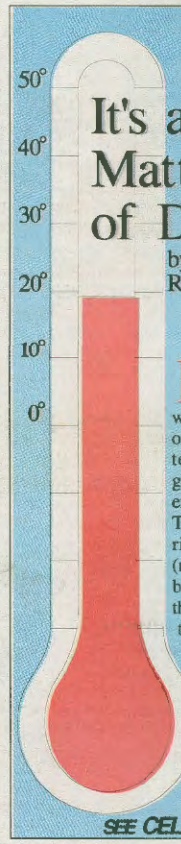
The assembly-language program shown in Listing 2 does all of this for us. The program starts with a RANDOM macro (adapted from my PUCHINKO program in THE RAINBOW, July 1991, Page 31) that assigns a random +1 or -1 change of direction to each bug. There are two 500-byte arrays that store all the necessary bug information. The LOC array stores the current *x* and *y* coordinates for each bug, and DEL holds the changes in direction for these coordinates. A bug's new location is its current location plus its current change in direction (no more than +1 or -1). Since each array is 500 bytes in length, there can be no more than 250 bugs.

After equating PSET, PPOINT and RND(0) to their actual ROM locations, NUMB holds the number of bugs, and LEN is the horizontal line with the imaginary bug zapper. The RANLOC routine picks random coordinates on the screen and checks to see if that point is already occupied (using PPOINT). If not, the coordinates are saved in the LOC array and that point is set (using) PSET. Next, the RANDOM macro assigns an initial change of direction of +1 or -1 to each *x* and *y* coordinate and stores the points in the DEL array.

Since most insects fly in a straight line for short distances, I've included a variable I refer to as Number Of Times. This counter (COUNT1) reflects how many times a bug may move in the same direction before new random changes are computed. I use a value of five, but you can make this as low as one, or higher if you want. Next, the current *x* and *y* coordinates are taken from the LOC array and that point is PRESET. Then the changes in the DEL array (500 bytes away) are added to the current coordinates. The program checks the *y* coordinate to keep it

SEE BUGGY ON PAGE 14

Feature Program



It's a Matter of Degree

by Roger Carlson

Every time I watch the weather forecasts on television, the temperatures are given in both Fahrenheit and Celsius. This aroused my curiosity somewhat (not so much why both were given but the relationship between the two). So I decided to write *Celsius*, a BASIC program designed to convert temperature readings from either system to the

SEE CELSIUS ON PAGE 20

OSK A User's Experiences

Feature Article

by John Donaldson

Like many CoCo 3 users, I wondered whether or not I should move to one of the OS-9/68000-based computers now available in this market. Having been a CoCo user since the days of the original Color Computer, I remember having only 4K of memory and using the original Color BASIC. From there I progressed to a 64K, OS-9 Level I system. And when the CoCo 3 came along, I bought one and soon found myself upgrading it to 512K of memory and adding OS-9 Level II and a hard drive.

When discussion of the so-called CoCo 4 machines started several years ago, I wrote to the three companies involved (IMS, FHL and Delmar) asking for all the information available. About the same time I joined a local Unix users group and gained access to USENET/INTERNET. Here I was able to ask others about using OS-9 on 68000-based computers. After getting all the facts and opinions I could, I decided to "move" to one of the new computers. I selected the MM/1 from Interactive Media Systems. In this article I share some of my experiences with OSK and the MM/1, and point out some differences between OSK and OS-9/6809.

SEE OSK ON PAGE 16

In this issue:

SEE PAGE 6

THE RAINBOW

Editor and Publisher
Lawrence C. Falk
Managing Editor Cray Augsburg
Associate Editor Sue Fomby
Submissions/Reviews Editor
Julie Hutchinson
Technical Editor Greg Law
Technical Assistants Ed Ellers
Contributing Editors Tim Kientzle, Martin Goodman, M.D., Eddie Kuns
Art Director Debbee Diamond
Designers Sharon Adams, Heidi Nelson
Consulting Editors Judi Hutchinson, Laurie D. Falk

Falsoft, Inc.

President Lawrence C. Falk

General Manager Peggy Lowry Daniels
Asst. General Mgr. for Finance
Donna Shuck
Admin. Asst. to the Publisher
Ellen Patterson
Editorial Director John Crawley
Director of Creative Services
O'Neil Arnold
Chief Bookkeeper/Dealer Accounts
Beverly Quick
Asst. Gen. Manager For Administration
Tim Whelan
Corporate Business Technical Director
Calvin Shields
Customer Service Manager
Beverly Bearden
Chief of Printing Services Melba Smith
Business Assistant Shannon Yoffe
Chief of Building Security and Maintenance
Lawrence Johnson
Advertising Assistant
Carolyn Fenwick
Western Sales Director
Ira Barsky
(312)587-1818
Eastern Sales Director
Kim Lewis
(502)228-4492

For RAINBOW Advertising and Marketing Office Information, see Page 23.

THE RAINBOW is published every month of the year by FALSOFT, Inc., The Falsoft Building, 9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059, phone (502) 228-4492. **THE RAINBOW**, **RAINBOWfest** and **THE RAINBOW** and **RAINBOWfest** logotypes are registered trademarks of FALSOFT, Inc. ● Second class postage paid Prospect, KY and additional offices. USPS N. 705-050 (ISSN No. 0746-4797). POSTMASTER: Send address changes to **THE RAINBOW**, P.O. Box 385, Prospect, KY 40059. Authorized as second class postage paid from Hamilton, Ontario by Canada Post, Ottawa, Ontario, Canada; GST No. R125434092. ● Entire contents copyright © by FALSOFT, Inc., 1992. **THE RAINBOW** is intended for the private use and pleasure of its subscribers and purchasers and reproduction by any means is prohibited. Use of information herein is for the single end use of purchasers and any other use is expressly prohibited. All programs herein are distributed in an "as is" basis, without warranty of any kind whatsoever. ● Tandy, Color BASIC, Extended Color BASIC and Program Pak are registered ®trademarks of the Tandy Corp. ● Subscriptions to **THE RAINBOW** are $31 per year in the United States. Canadian rates are U.S. $38. Surface mail to other countries is U.S. $68, air mail U.S. $103. All subscriptions begin with next available issue. ● Limited back issues are available. Please see notice for issues that are in print and their costs. Payment accepted by VISA, MasterCard, American Express, cash, check or money order in U.S. currency only. Full refund after mailing of one issue. A refund of 10/12ths the subscription amount after two issues are mailed. No refund after mailing of three or more magazines.

LETTERS

Needs Help With *Scripsit*

Editor:

I've been a subscriber to THE RAINBOW since February of this year. I bought my CoCo 3 primarily to see what I could learn to do with it, and the original Extended BASIC Manual has guided me most of the way. I'm very much a computer greenhorn, but I can see why CoCo fans like THE RAINBOW. I have a few questions for which I'd like answers.

Occasionally, when using *Scripsit* (Cat. No. 26-3105), the computer suddenly starts acting like the SHIFT key is being held down — it prints only capital letters. The only way I've found to stop this is to first save on tape whatever is in memory, then turn the CoCo off. When I turn it back on, I can reload from the tape and everything is back to normal.

Once while I was using BASIC, the CoCo began printing only lowercase characters on the screen. It would print capitals only when I used the SHIFT key. As before, I could return the computer to normal only by turning the CoCo off. What did I do wrong and can I correct it without having to turn the CoCo off?

Occasionally when loading files from tape, *Scripsit* will stop somewhere in the middle of the program and flash an error message on the screen. It might stop even in the middle of a word. I suspect a flaw in the tape, but is there any way I can make it resume loading the remaining part of the file?

Although the example for right justification in the DMP-107 manual (Page 35) works, how do I adapt it to work with a whole paragraph or more? Also, although both command examples work for printing bold and normal letters in the same line (Page 99) and italicized and non-italicized in the same line (Page 100), how do I tell the printer to print only specific words in a paragraph or sentence in bold or italics?

Is there any way to direct a BASIC commaud to the printer while I'm using *Scripsit*? And is there any way to load a program into *Scripsit* from tape that was CSAVED from within BASIC?

Is it true that all computers use the Microsoft BASIC language, except perhaps for the Atari and Commodore?

R.L. Aldrich
2505 Bernard
Denton, TX 76205

We suspect your problem with upper- and lowercase characters has to do with the SHIFT-Lock function of the CoCo keyboard. When you first turn on the CoCo, it is in the all-uppercase mode. You can switch to lowercase by holding down the SHIFT key and pressing 0 (zero). Pressing SHIFT-0 again returns you to the uppercase mode. When you start Scripsit, *it comes up in the lowercase mode. SHIFT-0 can be used to toggle in and out of the all-caps mode. Perhaps you were trying to enter a right parenthesis or asterisk and, while holding down SHIFT, pressed the 0 key by accident. If the problem occurs again, just try pressing SHIFT-0.*

Chances are, the error message you sometimes receive when loading files into Scripsit *is telling you there is a media error —a dropout or a bad spot on the tape. While you can't resume a stopped load, you can rewind and try again after adjusting the volume a little on the tape recorder.*

The BASIC interpreters for many personal computers were indeed written by Microsoft. This makes switching from one system to another a little easier. However, various personal computers offer different features, and Microsoft wrote these interpreters separately. Some commands available on a specific computer might not be available on others.

Hint for *VIP*/BASIC Editing

Editor:

I use *VIP Writer III* for editing BASIC programs, and I keep an ASCII copy of the program file on my *VIP* disk and a binary copy on my work disk. During a recent session, I loaded the ASCII copy of a program, made a few changes, then resaved it to my *VIP* disk. I then updated the copy on my work disk by saving the ASCII copy on top of the old binary copy.

However, when I attempted to run the program on my work disk, I received errors. Listing the program to the screen revealed that it had become garbled. It loaded into *VIP* just fine but was garbled whenever I tried to load it from BASIC.

When I checked the directory track, I discovered that *VIP* had not changed the file-type byte to reflect the fact that the copy on my work disk was an ASCII file. The BASIC interpreter was attempting to read tokens out of the ASCII text.

Therefore, when editing BASIC programs with *VIP*, you should never save an ASCII copy of a file on top of a binary copy. To avoid problems, save the ASCII copy with a different extension (such as .ASC) instead.

Trevor Boehm
77 Inwood Cres.
Winnipeg, MB R2Y 1A2
Canada

MIDI, MIDI Questions

Editor:

What does a MIDI keyboard do, and how do you hook it up to a CoCo? How does it enhance the sound capabilities of the CoCo (if that's what it does)? Which keyboard would you recommend in the $200-to-$300 price range.

Owen Crabtree
211 SW 6th Street
College Place, WA 99324

MIDI stands for Musical Instrument Digital Interface and is a standard that simplifies the interconnection of electronic musical instruments, as well as their use with computers. Most CoCo MIDI software is designed to send data to MIDI instruments though the rear serial port. Some products also support a special hardware pack once available from Speech Systems (perhaps Rulaford Research has a few). MIDI doesn't enhance *the sound capabilities of the CoCo; rather, it expands these capabilities using external hardware such as keyboards and drum machines. Managing editor Cray Augsburg has the Yamaha PSS-680 and also likes the Casio MT-540 and MT-240, though these units are not currently available new — check the bargain papers for used units. For more information about MIDI, see "Play It Again, CoCo" (December 1989, Page 74).*

Kudos to Coless

Editor:

I have a few good words to say about Mr. Walter Bayer and Coless Computer Design. I recently purchased *CIII PagesE 2.5*. Prior to my purchase, however, I wrote to Mr. Bayer with my concerns about printer support. I have a 512K CoCo 3, an RGB monitor and a DMP-130A printer, which according to the catalog isn't supported by *CIII PagesE*.

Not only did Mr. Bayer write back, he included a disk and a solution to the problem. Since then we have talked on the phone on a number of occasions, and Mr. Bayer has always been very helpful.

If you have a Tandy DMP-series printer and are looking for a good desktop publisher, give *CIII PagesE* a try. The product and the service is as advertised.

By the way, thank you for publishing an answer to my question in the August 1992 edition of "OS-9 Hotline." It was very helpful.

Ernest Bazzinotti, Jr.
91 Huggins Road
Rockland, MA 02370

Wants to Market BASIC Programs

Editor:

The first issue of THE RAINBOW I received was your last "glossy" issue. Although I liked that format better, I love THE RAINBOW anyway. I have written many BASIC programs that I would like to sell, but I can't afford to copyright and market them. Do you know of any companies that would buy them from me or pay royalties for them? What do you suggest?

Where can I get a Y-cable that I can use to connect my disk controller and modem (or Speech/Sound Cartridge) to my CoCo at the same time? And where can I get *CoCo Max III*? I have heard a lot about it but have never seen it advertised.

Is the OS-9 shell at the beginning of games like *Rogue*, *Microscopic Mission* and *Zone Runner* enough to run other programs that indicate "Requires OS-9 Level II?"

Leif Olson
238 4th NW
P.O. Box 124
Tioga, ND 58852

You can contact advertisers in THE RAINBOW to inquire about marketing your software through them. Better yet, send your programs to us—we're always interested in programs for publication.

We know of no company currently offering a Y cable. However, THE RAINBOW columnist Marty "The King of Kables" Goodman may be able to help. We have forwarded a copy of your letter to him.

For the most part, the OS-9 shell included with the games you mention is complete, and some third-party OS-9 offerings may work fine with it. However, this shell is minimal and doesn't include many of the utilities you really need to work with OS-9. There are several files in the OS9 Online databases on Delphi describing how to use these limited shells. OS-9 is available through Tandy's Express Order (call 1-800-321-3133).

Wants Program Paks

Editor:

I am writing to find out if you carry Program Paks for the Color Computer 2? If so, do you have a catalog of the games, and will you please send me a copy? I would also like to know the address for Radio Shack in Fort Worth.

David Basonic
804 East 18th Street
Cameron, TX 76520

No, we don't actually sell Program Paks, nor do we know of any CoCo vendors currently offering them commercially. Per-

haps other readers would part with theirs. Barring this, try Tandy's Express Order system at (800) 321-3133. You can write to Tandy/Radio Shack at One Tandy Center, Fort Worth, TX 76102.

High-Speed Disk Blues

Editor:

I am having trouble loading files on one of my disks. I was writing a program that runs in the high-speed mode and forgot to slow down the computer before I saved it. When I type DIR, I get an I/O error. I also get an I/O error whenever I try to use BASIC's DSKI$ command to see what the problem is. Do you know of any programming techniques with BASIC or BASIC09 that I can use to salvage the files on the disk? Where can I find programs that would remedy my situation?

Is there any way to alter the output of the RF port on a Commodore 64 so that it can be connected to an RGB monitor? If so, I would like to know how, no matter how complex the procedure might be. Please don't refer me to a past issue of THE RAINBOW — I don't have access to any of them. I would also like a ballpark figure on how much the parts will cost.

Also, I am interesting in finding a working 512K memory upgrade, a 1-meg memory upgrade, an FD-502 disk drive and working Color Computers (1, 2 or 3). I will try to answer all letters.

Brandon Broyles
4901 Wheeler Drive
The Colony, TX 75056

We fear your data is forever garbled. Once a file is saved at high speed, there is little chance of recovering it. For future reference, Steve Bjork wrote a utility called DFIX and uploaded it to the CoCo SIG on Delphi back when the CoCo 3 was first introduced. This machine-language routine allows high-speed disk access in most cases.

We have forwarded a copy of your letter to Marty "Monarch of Monitors" Goodman. Perhaps he can help you with your C64-to-RGB question.

A Bigger Binder for THE RAINBOW

Editor:

After reading "Print #-2" in the July 1992 issue, I decided to share the method I use to store THE RAINBOW. Radio Shack at one time carried (and can currently order) a 14⅞-by-11 printout binder (Cat. No. 26-211). If you punch two holes in the left margin of THE RAINBOW, the magazine fits in the binder just fine. And one binder holds 30 to 40 issues in the new format. Thank you for keeping THE RAINBOW alive!

Gary Erxleben
110 State Street
N. Pekin, IL 61554

THE RAINBOW welcomes letters to the editor. Mail should be addressed to: Letters to Rainbow, The Falsoft Building, 9509 U.S. Hwy 42, P.O. Box 385, Prospect, KY 40059. Letters should include the writer's full name and address. Letters may be edited for clarity or to conserve space.

Letters to the editor may also be sent to us through our Delphi CoCo SIG. From the CoCo SIG> prompt, enter RAI to get to the Rainbow Magazine Services area of the SIG. At the RAINBOW> prompt, enter LET to reach the LETTERS> prompt, then select Letters for Publication. Be sure to include your complete name and address.

CORRECTIONS

"PowerBoost: Speeding Up the CoCo" (Review, September 1992, Page 1): Needless to say, we are pretty excited that the 6309 from Hitachi works with the CoCo 3. Our enthusiasm, however, resulted in our misinterpreting part of Marty Goodman's review of Burke & Burke's PowerBoost. Contrary to what we reported, Burke & Burke *is not* at this time selling the 6309 microprocessor as a separate unit. It is, however, available as an optional part of PowerBoost. We apologize for any inconvenience our error may have caused.

"Sort Directories With BASIC09" (October 1992, Page 4): Author Ken Kobes contacted us to report a bug in *SortDir*. This error expands the directory file being sorted by one byte so that its length is no longer a multiple of 32. This creates a problem when OS-9 later expands the directory file to make room for new entries. To fix the bug, make the following corrections to SortDir.b09:

At Offset 00CF, change DIM zero:BYTE to DIM zero:STRING[32], and at Offset 00D6, change zero:=$00 to zero:=CHR$(0).

Submitting Material To Rainbow

Contributions to THE RAINBOW are welcome from everyone. We like to run a variety of programs that are useful, helpful and fun for other CoCo owners.

WHAT TO WRITE: We are interested in what you want to tell our readers. We accept for consideration anything that is well-written and has a practical application for the Tandy Color Computer. If it interests you, it will probably interest lots of others. However, we vastly prefer articles with accompanying programs that can be entered and run. The more unique the idea, the more the appeal. We have a continuing need for short articles with short listings. These are especially appealing to our many beginners.

FORMAT: Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs and debug our typing errors. All programs should be supported by some editorial commentary explaining how the program works. We also prefer that editorial copy be included in ASCII format on the tape or disk, using any of the word processors currently available for the Color Computer. Also, please include a double-spaced printout of your editorial material and program listing. Do not send text in all capital letters; use upper- and lowercase.

COMPENSATION: We do pay for submissions, based on a number of criteria. Those wishing remuneration should *so state* when making submissions.

For the benefit of those wanting more detailed information on making submissions, please send a self-addressed, stamped envelope (SASE) to: Submission Guidelines, THE RAINBOW, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit material currently submitted to another publication.

Feature Program

PASSWORD PROTECTION FOR THE COCO

by Joel Hegberg

It happens all the time. You're in the middle of a document or spreadsheet, or perhaps a game or other BASIC program, when you are interrupted and have to leave your CoCo unattended. If you live alone, this is no problem. But, for those of us who have been blessed with siblings or offspring who like to get into things they don't yet understand, coming back to a blank screen can be a most frustrating experience. To prevent this problem, I wrote *Password*, an assembly-language program designed to stop other would-be users from compounding your interruptions.

If you have an assembler (I used *Disk EDTASM+*), you can enter the source code in Listing 1 and assemble it. To execute the assembled program, you *must* enter

```
CLEAR 1000,32340:LOADM "PASSWORD
.BIN":EXEC 32340
```

This command line tells the CoCo that the highest address usable by BASIC is 32339, then loads and executes *Password*.

If you don't have an assembler or would prefer not to use one, enter the program shown in Listing 2 and save it to tape or disk. This program pokes the *Password* routine into the appropriate area of memory. Unlike other BASIC loaders, however, this program does not save the resultant machine-language program to tape or disk for you. Instead, it goes ahead and executes the routine.

The *Password* routine should be compatible with all BASIC programs and may also work with some machine-language software. To use it, just execute it when you turn your CoCo on, before running the other software you want to use. When you execute the program, you are instructed to enter a password of up to 10 characters. *Password* does not echo to the screen what you enter, so be careful when typing.

After your password is safely tucked away, *Password* multitasks with BASIC, scanning the keyboard roll-over table 60 times each second. To activate the routine, hold down the CTRL key and press L. A bell sounds to indicate *Password* is waiting for you to enter the correct password. To get back to what you were doing, simply type the password and press enter. Again, you will not be able to see what you are typing. If an incorrect password is entered, a long beep sounds, followed by another bell tone. If this happens, just try again — the process continues until the correct password is entered by the user.

It is important to know that while *Password* is active, *everything* is frozen. If the disk drive was on, it will stay on until BASIC regains control. Even the BASIC TIMER function stops! This can be a handy feature, especially if you're playing a game that bases your score on your speed — just use CTRL-L as a game-pause function.

Bit 0 CASSOUT
Bit 1 RS-232C output
Bit 2 6-Bit D/A
Bit 3 6-Bit D/A
Bit 4 6-Bit D/A
Bit 5 6-Bit D/A
Bit 6 6-Bit D/A
Bit 7 6-Bit D/A

Figure 1: Port $FF20 (PIA) Bit Assignments

Technical Considerations

Unlike some other machine-language programs, *Password* won't cause your printer to print garbage when the program produces its sounds. It is very irresponsible of programmers who overlook this bug, causing users to be startled as their printers jump to life for no apparent reason. It's also a waste of paper. The explanation of this bug is somewhat technical, but I think the problem is worth some discussion.

Memory Location $FF20 (65312 decimal) is the address which handles the CoCo's built-in 6-bit digital/analog converter. Among other things, this D/A converter is used to create sound. What many programmers don't realize is that this address also handles the CASSOUT line, which is used for cassette output (including controlling the Hi-Res joystick interface) and the RS-232C output line for the serial port. Figure 1 shows the bit assignments for Location $FF20.

In most cases, programs don't tell the computer to write to a cassette or scan joystick positions while producing sound. So the CASSOUT line usually doesn't present a problem. But if a printer is plugged in and online during sound output, a couple of "garbage" bits in the D/A converter can quickly leave a lasting impression — on paper.

Since you can always turn the printer off, this is a small price to pay to have high-quality sound on the CoCo. But the fact is, software is capable of filtering the garbage by using a single machine-language instruction, which I use in lines 780 and 980 of the assembly-language listing of *Password*. The program uses Register B to store the value it's about to send to the D/A converter. The command ANDB #252 tells the CoCo to ensure that the first two bits of the value in Register B are not set (they are zeros). Using this technique, even digitized sound can be filtered without any noticeable decrease in speed.

I urge all programmers to use filters in their sound routines. It does not affect the sound at all since the CASSOUT and RS-232C output lines are not connected to a speaker. Because sound filters are such an easy solution to an age-old CoCo problem, there is no excuse for not using them.

As a final note, *Password* doesn't protect against someone turning the CoCo off or pressing the Reset button. But it does keep the mildly nosy and unknowledgable from wreaking havoc while you are away from the computer. If you have any questions or comments about *Password*, feel free to write to me.

Joel Mathew Hegberg has been programming for nine years and enjoys writing software for the Color Computer and MM/1. Some of his commercial creations are available through Sub-Etha Software. Joel can be contacted at 936 N. 12th Street, De Kalb, IL 60115-2516, (815) 748-6638. Please include an SASE when requesting a reply.

32K Extended

Listing 1: PASSWORD

```
00100          ORG     32340
00110          TITLE   PASSWORD.ASM
00120
00130                          **************************
00140                          *"PASSWORD" WRITTEN FOR   *
00150                          *THE RAINBOW MAGAZINE BY  *
00160                          *JOEL MATHEW HEGBERG      *
00170                          *936 NORTH TWELFTH STREET *
00180                          *DE KALB, ILLINOIS  60115 *
00190                          **************************
00200
00210 BEGIN    LBRA    START
00220 DATA1    FCC     /ENTER YOUR PASSWORD./
00230          FCB     13
00240          FCB     0
00250 PASWRD   RMB     11
00260 BUFFER   RMB     11
00270 PRINT    PSHS    D,Y,U
00280 PRINT2   LDA     ,X+
00290          LBEQ    PRTDON
00300          JSR     [$A002]
00310          LBRA    PRINT2
00320 PRTDON   PULS    D,Y,U
00330          RTS
00340 START    LEAX    DATA1,PCR
00350          LBSR    PRINT
00360          LBSR    GETBUF
00370          LEAX    PASWRD,PCR
00380          LEAY    BUFFER,PCR
00390          LDB     #0
00400 LOOP1    LDA     ,Y+
00410          STA     ,X+
00420          INCB
00430          CMPB    #11
00440          LBLO    LOOP1
00450          LDX     269
00460          STX     IRQLOC,PCR
00470          LEAX    IRQ,PCR
00480          STX     269
00490          RTS
00500 IRQ      PSHS    D,X,Y,U
00510          LDA     342
00520          CMPA    #189
00530          LBEQ    FREEZ1
00540 IRQRTN   PULS    D,X,Y,U
00550          JMP     [IRQLOC,PCR]
00560 IRQLOC   RMB     2
00570 PATCH    PSHS    D,X,Y,U,CC
00580          ORCC    #80
00590          LDA     #189
00600          STA     342
00610          LDD     PRTLOC,PCR
00620          STD     360
00630          PULS    D,X,Y,U,CC
00640          JMP     [360]
00650 SNDON    LDA     $FF01
00660          ANDA    #247
00670          STA     $FF01
00680          LDA     $FF03
00690          ANDA    #247
00700          STA     $FF03
00710          LDA     $FF23
00720          ORA     #8
00730          STA     $FF23
00740          RTS
00750 TONE1    LBSR    SNDON
00760          CLRB
00770 TNLOP1   PSHS    B
00780          ANDB    #252    * SOUND FILTER
00790          STB     $FF20
00800          PULS    B
00810          COMB
00820          LDY     #80
00830 TNNXT1   EXG     D,Y
00840          SUBD    #1
00850          EXG     D,Y
00860          CMPY    #0
00870          LBNE    TNNXT1
00880          CMPB    #128
00890          LBHS    TNLOP1
00900          INCB
00910          CMPB    #128
00920          LBNE    TNLOP1
00930          RTS
00940 TONE2    LBSR    SNDON
00950          LDB     #20
00960          LDX     #200
00970 TNLOP2   PSHS    B
00980          ANDB    #252    * SOUND FILTER
00990          STB     $FF20
01000          PULS    B
01010          COMB
01020          LDY     #150
01030 TNNXT2   EXG     D,Y
01040          SUBD    #1
01050          EXG     D,Y
01060          CMPY    #0
01070          LBNE    TNNXT2
01080          EXG     D,X
01090          SUBD    #1
01100          EXG     D,X
01110          CMPX    #0
01120          LBNE    TNLOP2
01130          RTS
01140 PRTLOC   RMB     2
01150 FREEZ1   LDD     360
01160          STD     PRTLOC,PCR
01170          LEAX    PATCH,PCR
01180          STX     360
01190 FREEZE   LBSR    TONE1
01200          LBSR    GETBUF
01210          LEAX    BUFFER,PCR
01220          LEAY    PASWRD,PCR
01230          LDB     #11
01240 LOOP2    LDA     ,Y+
01250          LBEQ    PASS1
01260          CMPA    ,X+
01270          LBNE    NOPASS
01280          DECB
01290          LBEQ    PASSOK
01300          LBRA    LOOP2
01310 NOPASS   LBSR    TONE2
01320          LBRA    FREEZE
01330 PASS1    LDA     ,X
01340          LBNE    NOPASS
01350 PASSOK   LDD     PRTLOC,PCR
01360          STD     360
01370          LBRA    IRQRTN
01380 GETBUF   LDB     #0
01390          LEAX    BUFFER,PCR
01400 GETKEY   JSR     [$A000]
01410          LBEQ    GETKEY
01420          CMPA    #8
01430          LBEQ    BACKSP
01440          CMPA    #13
01450          LBEQ    ENTER
01460          CMPA    #32
01470          LBLO    GETKEY
01480          CMPA    #127
01490          LBHI    GETKEY
01500          CMPB    #11
01510          LBHS    GETKEY
01520          STA     ,X+
01530          INCB
01540          LBRA    GETKEY
01550 BACKSP   CMPB    #0
01560          LBEQ    GETKEY
01570          LDA     ,-X
01580          DECB
01590          LBRA    GETKEY
01600 ENTER    CMPB    #0
01610          LBEQ    GETKEY
01620          CMPB    #11
01630          LBEQ    CONT1
01640          CLR     ,X+
01650 CONT1    RTS
01660 DONE     EQU     *
01670          END
```

Listing 2: PASSWORD

```
10 'PASSWORD
20 'BY JOEL MATHEW HEGBERG
30 'COPYRIGHT (C) 1992
40 'BY FALSOFT, INC.
50 'RAINBOW MAGAZINE
60 '
70 'JOEL MATHEW HEGBERG
80 '936 NORTH TWELFTH STREET
90 'DE KALB, ILLINOIS 60115
100 '
110 CLEAR1000,32340:CLS:PRINT"PO
KING IN DATA.":PRINT"PLEASE WAIT
...";
120 LC=32340:RESTORE:LN=1000:TL=
0:LT=0
130 READ A$:IFA$="**"THEN200
140 IFLEN(A$)>2THEN170
150 V=VAL("&H"+A$):POKE LC,V:LC=
LC+1:LT=LT+V
160 GOTO 130
170 V=VAL(A$):IF V<>LT THEN PRIN
T"DATA ERROR IN LINE #";LN:STOP
180 TL=TL+LT:LT=0:LN=LN+10
190 GOTO 130
200 READ A$:V=VAL(A$):TL=TL+LT
210 IF V<>TL THEN PRINT"UNLOCATA
BLE ERROR IN DATA STATEMENTS.":S
TOP
220 PRINT:PRINT:EXEC32340
230 NEW
1000 DATA 16,0,3E,45,4E,54,45,52
,20,59,4F,55,52,20,50,41,53,53,1
176
1010 DATA 57,4F,52,44,2E,D,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3
75
1020 DATA 0,0,0,0,0,34,66,A6,80,
10,27,0,7,AD,9F,A0,2,16,FF,F3,15
24
1030 DATA 35,66,39,30,8C,BF,17,F
F,E8,17,1,11,30,8C,CC,31,8C,D4,1
935
1040 DATA C6,0,A6,A0,A7,80,5C,C1
,B,10,25,FF,F5,BF,1,D,AF,8D,0,21
88
1050 DATA 19,30,8D,0,4,BF,1,D,39
,34,76,B6,1,56,81,BD,10,27,0,9C,
1448
1060 DATA 35,76,6E,9D,0,0,0,0,34
,77,1A,50,86,BD,B7,1,56,EC,8D,16
85
1070 DATA 0,85,FD,1,68,35,77,6E,
9F,1,68,B6,FF,1,84,F7,87,FF,1,22
93
1080 DATA B6,FF,3,84,F7,87,FF,3,
B6,FF,23,8A,8,87,FF,23,39,17,FF,
2686
1090 DATA E4,5F,34,4,C4,FC,F7,FF
,20,35,4,53,10,8E,0,50,1E,2,83,1
902
1100 DATA 0,1,1E,2,10,8C,0,0,10,
26,FF,F1,C1,80,10,24,FF,DD,5C,16
80
1110 DATA C1,80,10,26,FF,D6,39,1
7,FF,B5,C6,14,8E,0,C8,34,4,C4,21
72
1120 DATA FC,F7,FF,20,35,4,53,10
,8E,0,96,1E,2,83,0,1,1E,2,10,144
6
1130 DATA 8C,0,0,10,26,FF,F1,1E,
1,83,0,1,1E,1,8C,0,0,10,26,FF,13
33
1140 DATA D5,39,0,0,FC,1,68,ED,8
D,FF,F7,30,8D,FF,61,8F,1,68,17,2
367
1150 DATA FF,8A,17,0,34,30,8D,FF
,F9,31,8D,FE,EA,C6,B,A6,A0,10,23
89
1160 DATA 27,0,14,A1,80,10,26,0,
8,5A,10,27,0,F,16,FF,EC,17,FF,13
61
1170 DATA 95,16,FF,D6,A6,84,10,2
6,FF,F4,EC,8D,FF,BC,FD,1,68,16,2
691
1180 DATA FF,1C,C6,0,30,8D,FE,C3
,AD,9F,A0,0,10,27,FF,FB,81,8,230
6
1190 DATA 10,27,0,1E,81,D,10,27,
0,24,81,20,10,25,FF,E6,81,7F,127
3
1200 DATA 10,22,FF,E0,C1,B,10,24
,FF,DA,A7,80,5C,16,FF,D4,C1,0,23
27
1210 DATA 10,27,FF,CE,A6,82,5A,1
6,FF,C8,C1,0,10,27,FF,C2,C1,8,22
80
1220 DATA 10,27,0,2,6F,80,39,0,*
*,41190
```

CBASIC Editor/Compiler
The ULTIMATE Color Computer Basic Compiler!!!

If you want to write fast machine language programs but you don't want to spend the next few years trying to write them in Assembly Language , then CBASIC is the answer!!!

CBASIC is the only fully integrated Basic Compiler and program editing system available for the Color Computer. It will allow you to take full advantage of all the capabilities available in your color computer without having to spend years trying to learn assembly language programming. CBASIC allows you to create, edit and convert programs from a language you are already familiar with Extended Disk Color Basic, into fast efficient machine language programs easily and quickly.

"The most complete Editor/Compiler I have seen for the CoCo... "-The RAINBOW March 1986

CBASIC is a powerful tool for the Beginner as well as the Advanced Basic or M.L. programmer. CBASIC features well over 150 compiled Basic Commands and Functions that fully support Disk Sequential and Direct access files, Tape, Printer and Screen I/O. CBASIC supports ALL the High and Low Resolution Graphics, Sound, Play and String Operations available in Enhanced Color Basic, including Graphics H/GET, H/PUT, PLAY and H/DRAW, all with 99% syntax compatibility.

Specify Coco 1, 2 or 3 Disk $149.00

DataPack III Plus V1.1

AUTOPILOT and AUTO-LOG Processors
X-MODEM DIRECT DISK FILE TRANSFER
VT-100 & VT-52 TERMINAL EMULATION

- No lost data even at 2400 Baud on the Serial I/O port.
- 8 Selectable Display Formats, 32/40/64/80 columns
- ASCII & BINARY disk file transfer via XMODEM.
- Directly record receive data (Data Logging).
- VT-100 emulation for VAX, UNIX and other systems.
- VT-100/52 cursor keys, position, PF & Alt. Kbd. keys.
- Programmable Word Length, Parity, Stop Bits
- Complete Full and Half Duplex operation,
- Send full 128 character set from Keyboard .
- Complete Editor, Insert, Delete, Change or Add .
- 9 Variable length, Programmable Macro Key buffers.
- Programmable Printer rates from 110 to 9600 Baud.
- Send Files from the Buffer, Macro Key Buffers or Disk.
- Display on Screen or Print the contents of the Buffer.
- Freeze Display & Review information On line .
- Built in Command Menu (Help) Display.

Supports: Modem-Pak & Deluxe Pak or Serial Port.

Specify Coco 1, 2, 3 Disk $49.95

EDT/ASM III

EDT/ASM III is a Disk based co-resident Text Editor & Assembler. It is designed to take advantage of the new features of the COCO 3. It has 8 Display formats from 32/40/64/80 columns. The disk also contains a free standing ML Debug Monitor.

EDT/ASM III has the most powerful, easy to use Text Editor available in and Editor/Assembler package for the Color Computer.

* Local and Global string search and/or replace.
* Full Screen line editing.
* Load and Save standard ASCII formatted files.
* Block Moce & Copy, Insert, Delete, Overtype.
* Create and Edit files larger than memory.

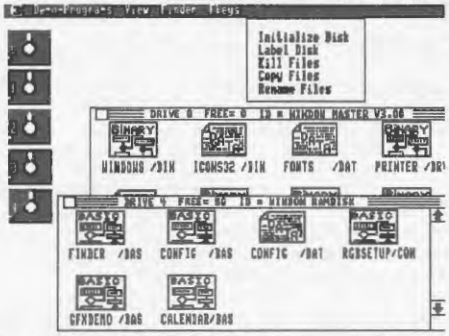
The Assembler features include:

* Conditional If/Then/Else assembly.
* Disk Library files up to 9 levels deep.
* Supports standard Motorola directives.
* Allows multiple values in FCB/FDB directives.
* Allows assembly from Buffer, Disk or both.

Specify Coco 1, 2 or 3 Disk $59.95

CER COMP

Window Master V3.0



The Ultimate User Friendly Point & Click operating System for your Coco 3. Simple enough even for children to use, just point and click to run programs, select files, do disk or file maintenance or almost any task you currently do by typing commands. You also get things like a print spooler, Programmable Function Keys, a Buffered Keyboard, Ramdisk, Serial I/O port and Deluxe Pak support along with Windows, Icons, Buttons, Pull Down Menus, Edit Fields and Mouse functions all in one program. It has multiple fonts in 54 possible sizes and styles, Enhanced Basic Editing and much much more. It add over 50 commands and Functions to Basic to fully support the Point & Click System without OS9.

$49.95

"..it offers so many features that it is probably underpriced. I recommend this software to all CoCo3 owners." -The Rainbow Feburary 1989

It is completely compatible with existing Basic programs and takes absolutely no memory away from basic. It requires 1 Disk Drive, R.S. hires interface & Joystick or Mouse.

Includes 128 & 512K Versions Only $69.95

LOOK!

"The Source"

The SOURCE will allow you to easily and quickly Disassemble machine language programs directly from disk and generate beautiful, Assembler Source Code. And " The Source" has all the features and functions you are looking for in a Disassembler .

* Automatic Label generation.
* Allows specifying FCB, FCC and FDB areas.
* Disassembles programs directly from Disk.
* Automatically locates addresses.
* Output listing to the Printer, Screen or both.
* Generates Assembler source directly to disk.
* Built in Hex/ASCII dump/display.
* 8 Selectable Display formats 32/40/60/80.
* Built in Disk Directory and Kill file commands.
* Menu display with single key commands.
* Written in Ultra fast machine language.

Specify Coco 1, 2 or 3 Disk $49.95

Window - Ware

Window Writer- A point & click Word Processor, powerful formatting capability, works wi...th any printer. On screen Italic, bold etc. WYSIWYG..... ..... $59.95

"Window Writer is a powerful word processor that is fun to use, very user friendly..... The Enviornment compares favorably to that of Microsoft Windows " - RAINBOW Dec. 1989

Window Writer/W- for non W/M owners.....$79.95
Window Basic Compiler-similar to CBASIC...$99.00
Window Edt/Asm- A full featured Assembler...$49.95
Font/Icon Editors- Basic & M.L. versions..... $19.95
Advanced Programmers Guide- $24.95
The Memory game- Concentration game $19.95
Desk Accessory Pak-7 resident programs $39.95

Deluxe Terminal

A Completely New and Easy to use Terminal Program designed specifically for the Coco III.

With advanced features you would expect to find only in a Hi-Priced MS-Dos program. It has a 26 Entry Phone Directory with complete Configuration information for Communications and Automatic logon. Supports the Serial I/O Port up to 2400 baud, Deluxe & Modem Pak and the Disto Serial I/O board up to 9600 baud. It has a Full Screen Text Editor, X/Y Modem File transfer support, Split Screen Conference Mode, Macro Keys. Full Disk Support including Multifile Copy, Kill, Rename, Arc/Un-Arc and Disk Initialize and it is Completely Compatible with ADOS.

Requires 128K, Disk & 80 Col. Display $44.95

CoCo III Tool Kit

Disk Commands
Backup, Initialize, Directory, Verify, Compare, Search, Edit, Erase, Speed Test, Step Rate Test, Gran Table Analysis & Repair

File Commands
Arcive, Copy, Kill, Rename, Erase, View, Edit, Print, Compare, Salvage, Search, Verify, Test Arcive, Un-Arcive, Xmodem Send/Rec.

The Disk Utility Program that you always dreamed of is now a Reality.

Coco Tools is a comprehensive disk utility Program providing the most complete set of functions available for the standard R.S. DOS disk system. Comparable in scope and functionality to that of the famous utility available for MS-DOS computers "PC-TOOLS"!

Coco Tools is also the most Comprehensive Disk Repair program available, it Automatically diagnosis and repairs file allocation errors, locates corrupt directory information and cross linked files. It provides fast and easy recovery of deleted files without the drudgery normally associated with killed file recovery.

Requires 128K, Disk & 80 col. display $49.95

This Months Specials
Savings Up To 35%

EDT/ASM III $49.95
Window Master.....$49.95
Window Master & Window Writer.....$79.95
Deluxe Terminal.....$34.95

VISA, MASTERCARD or COD

Call Toll Free 1-800-383-8529

(Monday thru Saturday, 8am to 5pm PST).

To order by mail, send check or money order for the amount of the program plus $4.00 for shipping to:

Cer-Comp Ltd.
5566 Ricochet Avenue
Las Vegas, NV 89110
702-452-0632

Feature Program

MINEFIELD

by KENNETH REIGHARD JR.

Cautiously you creep through the freshly turned field. Your detector indicates mines nearby, but in which direction do they lie? The tension mounts with each slow step. Suddenly you are overcome with fear, and you run in a random direction. Then . . . KABOOM!

Minefield is a logic game for the CoCo 3. As written, the program requires a disk drive, though I'll show you in a moment how to modify it for tape-based systems.

Before you run *Minefield* the first time, you must create the data file (BESTMINE.DAT) in which it stores the best playing time. Do this by entering RUN 570. You can also use this method to reset the times stored in the file. Once BESTMINE.DAT exists on your playing disk, you can run the program simply by entering RUN "MINFIELD". When you run the program, you are prompted for whether you have a composite (press C) or RGB (press R) monitor. If you are using a television, use the Composite option.

Minefield supports three skill levels: Easy is played on a 10-by-10 grid with 10 mines; Medium is played on a 20-by-15 grid and uses 45 mines; and Hard is played on a 30-by-20 grid hiding 99 mines. Select the desired skill level by pressing 1, 2 or 3 on the title screen.

When you first start the game, all the squares on the grid (minefield) are covered. The object of the game is to uncover all the unmined squares without "stepping" on a mine. The cursor indicating the current square appears in the upper-left corner; use the arrow keys to move the cursor to the various squares on the grid.

To place a flag on a square you think contains a mine, simply move the cursor to that square and press ENTER. The mine counter in the upper-right corner of the screen will decrease by one. If you later change your mind, you can remove the flag by again selecting that square and pressing ENTER. It is important that you understand the mine counter is based on the *number* of flags you place and has nothing at all to do with whether or not mines actually exist at their locations. You cannot place more markers than the number of mines on any given level.

To uncover a square, select it with the cursor and press the space bar. If the square contains a mine, well . . . let's just say the game is over. If this happens, all hidden mines are revealed and any flags you have placed inaccurately are indicated with an X. If you are lucky and don't step on a mine when you press the space bar, you will see either a blank space or a number. A blank indicates there are no mines hidden in any of the eight adjacent squares. (Yes, diagonals count in a minefield.) At this point the game reveals all blank squares adjacent to the selected square in all directions until number squares are reached. A number indicates the number of adjacent squares that contain mines. For example, if a square contains a 2, you know there two mines next to it.

When you uncover all the squares that don't contain mines, your playing time is displayed. Scoring is based on the time it takes you to solve an entire minefield. If your time is the best score for the current playing level, you are asked to enter your name. *Minefield* is designed to save the best time from each level in the BESTMINE.DAT file on disk. This routine must be sacrificed if you are using a tape-based system. To remove the routine, delete lines 485 through 495, 510 through 550, and 570 through 585.

Remember that *Minefield* is for the most part a logic game. As tempted as you might become, try to guess as little as possible. However, there are some possible number combinations on the grid that will force you to guess at times. After all, luck has a little to do with it in real life, too.

Ken Reighard Jr. is studying computer science and engineering at the University of Toledo, where he is also a member of the Triangle fraternity. He can be contacted at 2F441 Ridgeland Drive, Toronto, OH 43964, (614) 537-4875. Please include an SASE when requesting a reply.

Arrows	move cursor around grid
ENTER	a) set flag for possible mine location
	b) remove existing flag marker
Space Bar	uncover selected square

***Figure 1:** Minefield* **Key Commands**

CoCo 3 Disk

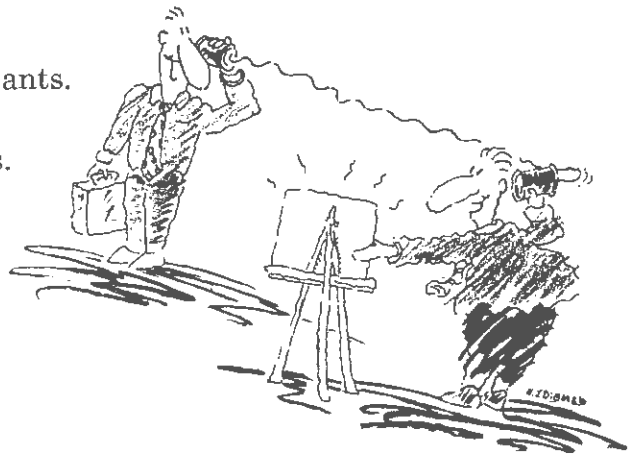
The Listing: MINFIELD

```
0 'MINEFIELD
1 'BY KENNETH REIGHARD, JR.
2 'COPYRIGHT (C) 1992
3 'BY FALSOFT, INC.
4 'RAINBOW MAGAZINE
5 GOTO 10
6 HPAINT(H*8+16,V*8+16),0,2:RETU
RN
7 HPAINT(H*8+16,V*8+16),4,2:RETU
RN
8 HPAINT(H*8+16,V*8+16),3,2:RETU
RN
9 HPAINT(H*8+16,V*8+16),1,2:RETU
RN
10 CLS:INPUT"MONITOR (R/C)";Q$
15 IF Q$="C" THEN PALETTE CMP EL
SE PALETTE RGB
20 POKE 65497,0
25 DIM M(29,19), N(29,19), N$(3)
, T(3), SX(100), SY(100)
30 WIDTH40:CLS3:ATTR 3,2
35 HCOLOR 3,8:HSCREEN 2
40 HPRINT(15,1),"MINEFIELD!":HPR
INT(8,3),"By Kenneth Reighard, J
r."
45 HPRINT(16,6),"1. Easy":HPRINT
(16,7),"2. Medium":HPRINT(16,8),
"3. Hard"
50 Q$=INKEY$:SK=VAL(Q$):IF SK<1
OR SK>3 THEN 50
55 HPRINT(10,20),"One Moment Ple
ase..."
60 ON SK GOTO 65,70,75
65 MX=9:MY=9:B=10:GOTO 80
70 MX=19:MY=14:B=45:GOTO 80
75 MX=29:MY=19:B=99
80 L=B
85 F$="C8BD6BR4U5FL":B$="C8BD6BR
4H2E2F2G2U4"
90 R=(MX+1)*(MY+1)-B
95 H=RND(-TIMER)
100 FOR Z=1 TO B
105 H=RND(MX+1)-1:V=RND(MY+1)-1
110 IF M(H,V)=-1 THEN 105 ELSE M
(H,V)=-1
115 NEXT Z
120 FOR V=0 TO MY:FOR H=0 TO MX
125 IF M(H,V)<>-1 THEN 155
130 FOR Y=-1 TO 1:FOR X=-1 TO 1
135 IF X=0 AND Y=0 THEN 150
140 IF H+X<0 OR H+X>MX OR V+Y<0
OR V+Y>MY THEN 150
145 IF M(H+X,V+Y)<>-1 THEN M(H+X
,V+Y)=M(H+X,V+Y)+1
150 NEXT X,Y
155 NEXT H,V
160 HCOLOR 2,8:HCLS
165 HPRINT(34,0),"Mines"
170 HLINE(15,15)-(MX*8+23,MY*8+2
3),PSET,B
175 HPAINT(16,16),4,2
180 FOR X=0 TO MX
185 HLINE(X*8+15,15)-(X*8+15,23+
MY*8),PSET
190 NEXT X
195 FOR Y=0 TO MY
200 HLINE(15,Y*8+15)-(23+MX*8,Y*
8+15),PSET
205 NEXT Y
210 H=0:V=0:T=0
215 TIMER=0
220 HCOLOR 7:HLINE(H*8+15,V*8+15
)-(H*8+23,V*8+23),PSET,B
225 IF R=0 THEN 440
230 T=T+TIMER:TIMER=0
235 HCOLOR 2:HLINE(288,8)-(304,1
6),PRESET,BF:HPRINT(35,1),L
240 Q$=INKEY$:IF Q$="" THEN 240
245 HCOLOR 2:HLINE(H*8+15,V*8+15
)-(H*8+23,V*8+23),PSET,B
250 IF Q$=CHR$(8) THEN H=H-1:IF
H<0 THEN H=MX:GOTO 220 ELSE 220
255 IF Q$=CHR$(9) THEN H=H+1:IF
H>MX THEN H=0:GOTO 220 ELSE 220
260 IF Q$=CHR$(10) THEN V=V+1:IF
 V>MY THEN V=0:GOTO 220 ELSE 220
265 IF Q$=CHR$(94) THEN V=V-1:IF
 V<0 THEN V=MY:GOTO 220 ELSE 220
270 IF Q$=CHR$(13) THEN IF N(H,V
)=0 AND L>0 THEN GOSUB 9:HDRAW"B
M"+STR$(H*8+15)+","+STR$(V*8+15)
+F$:N(H,V)=1:L=L-1:GOTO 220 ELSE
 IF N(H,V)=1 THEN GOSUB 7:N(H,V)
=0:L=L+1:GOTO 220 ELSE 220
275 IF Q$<>CHR$(32) THEN 220
280 IF N(H,V)<>0 THEN 220
285 C=M(H,V)
290 GOSUB 6
295 IF C=-1 THEN 400
300 R=R-1
305 IF C=0 THEN N(H,V)=3:GOTO 32
5
310 HCOLOR 3:HPRINT(H+2,V+2),CHR
$(C+48)
315 N(H,V)=2
320 GOTO 220
325 OH=H:OV=V
330 ST=1:SX(ST)=H:SY(ST)=V
335 IF ST=0 THEN H=OH:V=OV:GOTO
220
340 X=SX(ST):Y=SY(ST)
345 FOR H=X-1 TO X+1:FOR V=Y-1 T
O Y+1
350 IF H<0 OR H>MX OR V<0 OR V>M
Y THEN 390
355 IF N(H,V)<>0 THEN 390
360 IF M(H,V)>0 THEN N(H,V)=2:R=
R-1:GOSUB 6:HCOLOR 3:HPRINT(H+2,
V+2),CHR$(M(H,V)+48):GOTO 390
365 IF M(H,V)=-1 THEN 390
370 GOSUB 6
375 N(H,V)=3
380 R=R-1
385 ST=ST+1:SX(ST)=H:SY(ST)=V:GO
TO 340
390 NEXT V,H
395 ST=ST-1:GOTO 335
400 GOSUB 8:HDRAW"BM"+STR$(H*8+1
5)+","+STR$(V*8+15)+B$:PLAY"O1L2
55V31CDEFGAFGFDCGFV15"
405 FOR V=0 TO MY:FOR H=0 TO MX
410 IF M(H,V)=-1 THEN GOSUB 8:HD
RAW"BM"+STR$(H*8+15)+","+STR$(V*
8+15)+B$
415 IF N(H,V)=1 AND M(H,V)<>-1 T
HEN GOSUB 9:HDRAW"C3BM"+STR$(H*8
+15)+","+STR$(V*8+15)+"F8BU8G8"
420 NEXT H,V
425 HCOLOR 2:HPRINT(14,23),"Pres
s Key..."
430 Q$=INKEY$:IF Q$="" THEN 430
435 GOTO 480
440 T=T+TIMER:T=INT(T/60)
445 HCOLOR2:HPRINT(15,22),"Time=
"+STR$(T):HLINE(H*8+15,V*8+15)-(
H*8+23,V*8+23),PSET,B
450 FOR V=0 TO MY:FOR H=0 TO MX
455 IF M(H,V)=-1 AND N(H,V)<>1 T
HEN GOSUB 9:HDRAW"BM"+STR$(H*8+1
5)+","+STR$(V*8+15)+F$
460 NEXT H,V
465 PLAY"O2L4CEFL2GL4DFGL203CP2
470 HCOLOR 2:HPRINT(14,23),"Pres
s Key..."
475 Q$=INKEY$:IF Q$="" THEN 475
480 POKE65496,0
485 OPEN "I",#1,"BESTMINE.DAT"
490 INPUT #1,N$(1),T(1),N$(2),T(
2),N$(3),T(3)
495 CLOSE #1
500 HSCREEN 0
505 CLS
510 NS=0:IF R=0 THEN IF T<T(SK)
THEN PRINT"You've beat the best
time!":LINEINPUT"Enter your name
 :";N$:T(SK)=T:N$(SK)=N$:NS=1
515 PRINT:PRINT"Best Times":PRIN
TSTRING$(40,42);
520 PRINTUSING"Easy      #### ";T
(1);:PRINTN$(1)
525 PRINTUSING"Medium    #### ";T
(2);:PRINTN$(2)
530 PRINTUSING"Hard      #### ";T
(3);:PRINTN$(3)
535 IF NS=0 THEN 555
540 OPEN "O",#1,"BESTMINE.DAT"
545 WRITE#1,N$(1),T(1),N$(2),T(2
),N$(3),T(3)
550 CLOSE#1
555 PRINTSTRING$(40,42);:PRINT:P
RINT"Play Again (Y/N)?"
560 Q$=INKEY$:IF Q$="Y" THEN RUN
 20 ELSE IF Q$="N" THEN END ELSE
 560
565 END
570 OPEN "O",#1,"BESTMINE.DAT"
575 WRITE#1,"",9999,"",9999,"",9
999
580 CLOSE#1
585 END
```

DELPHI – The $1 per hour online solution!

DELPHI's 20/20 Advantage Plan sets the standard for online value: 20 hours for only $20, for all the services you want!

- Thousands of files to download.
- Chat lines with hundreds of participants.
- Worldwide e-mail.
- Hobby and computer support groups.
- Multi-player games.
- Local access numbers in over 600 cities and towns.



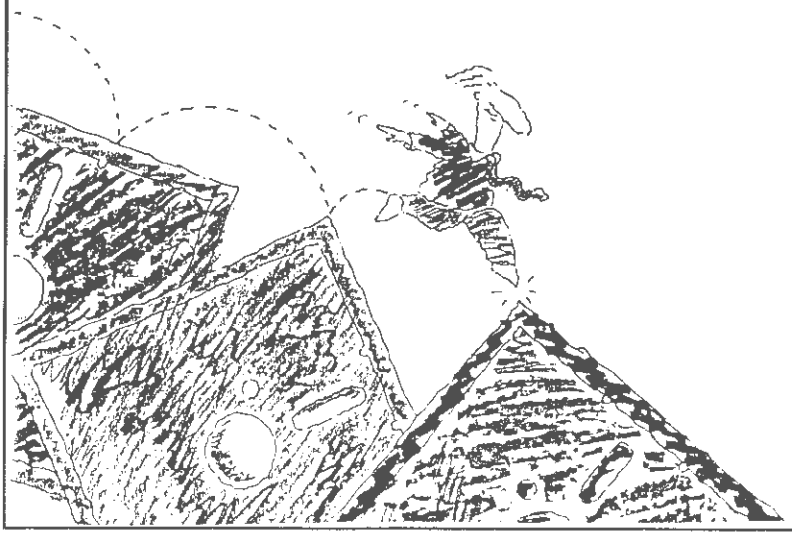
Trial Offer: 5 hours for $5!

Try DELPHI at $1 per hour. Join today and get 5 hours of evening and weekend access for only $5. If you're not satisfied, simply cancel your account before the end of the calendar month with no further obligation. Keep your account active and you'll automatically be enrolled in the 20/20 plan for the next month.

1. Via modem, dial 1-800-365-4636.
2. When connected, press RETURN once or twice.
3. At *Password*, enter **RB55**

Questions? Call 1-800-695-4005.

Rates apply for evening and weekend access from within the mainland US. There is a one-time enrollment fee of $19 when you join the 20/20 Advantage Plan. Further details are provided during the online registration.



DELPHI is a service of
General Videotex Corporation
1030 Massachusetts Avenue
Cambridge, MA 02138-5302

800-695-4005 • 617-491-3393

TIM KIENTZLE

A Hexadecimal Trick

Some years ago I came across a trick for converting hexadecimal numbers into ASCII by taking advantage of binary-coded decimal numbers. I'll show you this simple trick and explain how it works. Let's start by looking at the problem.

A short segment from the standard ASCII chart appears in Figure 1. To convert a value from $0 through $9 to its ASCII value, we can simply add $30 to get the correct ASCII code ($30 through $39). So, our first attempt at an assembly-language Hex-to-ASCII conversion is the single 6809 instruction:

```
adda #$30
```

However, this doesn't work for numeric values higher than $9. We can almost get around this by thinking in terms of *binary-coded decimal* (BCD) numbers. In BCD, each digit from the decimal number is represented by four bits (a nibble). For example, 78_{10} (the subscript 10 indicates a base of 10 — a decimal number) is represented in BCD as 0111 1000 since the binary nibble 0111 is the same as decimal 7 and 1000 is decimal 8. (Notice that in straight binary conversion, 78_{10} becomes 01001110_2.) Since two nibbles (representing two decimal digits) conveniently fit into one byte, BCD numbers are often stored in this form. A chart of the decimal digits and their BCD equivalents is shown in Figure 2.

ASCII	Hexadecimal
0	$30
1	$31
2	$32
3	$33
4	$34
5	$35
6	$36
7	$37
8	$38
9	$39
@	$40
A	$41
B	$42
C	$43
D	$44
E	$45
F	$46

Figure 1: ASCII Values

Decimal	BCD
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
n/a	1010
n/a	1011
n/a	1100
n/a	1101
n/a	1110
n/a	1111

Figure 2: Decimal/BCD Equivalents

Looking again at Figure 1, we see that adding $30 to 10 (the decimal value of Hex A) using BCD addition gives us $40, which is *almost* the ASCII code for the character A. So our second attempt at conversion is two instructions:

```
adda   #$30
daa
```

The `daa` instruction (which stands for *decimal adjust accumulator*) is used to convert a normal addition in Register A into a BCD result. It can be used only after an addition or subtraction instruction since it relies on the value of the half-carry flag. (This flag is set only by addition and subtraction instructions, so the `daa` instruction can't be used to simply convert a number to BCD.)

This second attempt results in a conversion that is different only by a value of one for Hex values above 10. If we could find some way to set the carry for values above nine, we could use an "addition with carry" to get the right answer. A compare instruction such as `CMPA #10` will set the carry for values below 10, which is exactly the opposite of what we want. So we'll use another trick with BCD arithmetic:

```
adda   #$90
daa
```

This generates a carry for values higher than nine. Combining this with the idea above, and compensating for the addition of $90, we end up with the following code segment:

```
adda   #$90
daa
adca   #$40
daa
```

This short segment correctly converts any single hexadecimal digit into its ASCII value. Using this, we can write a short routine to convert a value in Register A into two ASCII digits in registers A and B. One version of the resulting routine is shown in Figure 3.

```
hex2asc  tfr    a,b     Make a copy of A in B
         anda   #$0F    Mask off lower hex digit
         adda   #$90    Convert digit to ASCII
         daa
         adca   #$40
         daa
         exg    a,b     Move ASCII to B, restore A
         lsra
         lsra
         lsra
         lsra           Select upper hex digit
         adda   #$90
         daa
         adca   #$40
         daa
         rts            Return two bytes of ASCII
```

Figure 3: Hex-to-ASCII Routine

Not all programmers use the same pattern for naming variables and data structures. Even one programmer uses different approaches at different times. Next time we'll look at how your choices affect binary housekeeping.

Tim Kientzle is currently pursuing a doctorate in mathematics at the University of California at Berkeley. He is the author of V-Term *and has worked with the Color Computer since 1982.*

Feature Program

Queues

Keep You In Line
Keep You In Line
Keep You In Line

by Joseph Pendell

Xqueue illustrates a simple use for queues in BASIC programming. The program first draws an X in the middle of the screen. It then divides the screen into four sections, in each of which it draws another X. These X's further divide the screen and more X's are drawn, and the process continues.

The way *Xqueue* does this is by using a *queue*. A queue is merely a list to which items are added at one end and taken off from the other. This is similar to a line at the bank in which people start at the end of the line, and the teller waits on the person in the front. *Xqueue* uses two arrays, `A` and `B`, to hold the screen coordinates for the center of each X added to the queue.

Variable `M` in Line 20 controls how far from the middle X each of the smaller X's are drawn. Variable `R` in the same line reflects the rate at which the size of the X's decreases. These two variables are the best to change in order to produce different-looking results. Variable `L` used in lines 70 and 80 represents the length of the current X. Throughout *Xqueue*, `I` is used to indicate the location in the arrays that contains the coordinates for the next X to be drawn. Similarly, `N` indicates the location in the arrays where the next coordinates can be put in the queue.

Unless you use your imagination, *Xqueue* is a "do-nothing" program — though it does draw a fairly interesting design. With a little thought, however, you can put queues to work in your programming efforts.

Joseph Pendell has a degree in electrical engineering from the University of Maryland. In addition to programming the Color Computer, Joseph enjoys using the Macintosh. His hobbies include riding skateboards and playing Super Nintendo.

16K Extended

The Listing: XDRAW

```
1 'XDRAW
2 'BY JOSEPH PENDELL
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
20 M=.8:R=3.3
30 DIM A(90),B(90)
40 A(0)=128:B(0)=96:L=80
50 I=0:N=1:S1=1:S2=1
60 PMODE 4,1:PCLS:SCREEN1,1
70 LINE(A(I)-L,B(I)-L)-(A(I)+L,B
(I)+L),PSET
80 LINE(A(I)-L,B(I)+L)-(A(I)+L,B
(I)-L),PSET
90 LS=M*L
100 'STORE 4 NEW LOCATIONS
110 A(N)=A(I):B(N)=B(I)-LS
120 A(N+1)=A(I)+LS:B(N+1)=B(I)
130 A(N+2)=A(I):B(N+2)=B(I)+LS
140 A(N+3)=A(I)-LS:B(N+3)=B(I)
150 N=N+4
160 S2=S2+1:IF S2>S1 THEN S1=S1*
4:L=L/R:S2=1
170 I=I+1:IF N<85 THEN 70
180 'DRAW THE LAST ROUND OF X'S
190 FORJ=I TO (N-1)
200 LINE(A(J)-L,B(J)-L)-(A(J)+L,
B(J)+L),PSET
210 LINE(A(J)-L,B(J)+L)-(A(J)+L,
B(J)-L),PSET
220 NEXTJ
230 GOTO230
```

OWL-WARE

Proven Technology

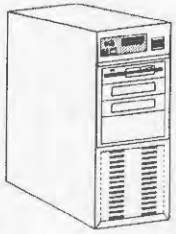
On the Razor's Edge of the Color Computer Frontier

In our 10th Year !

A DECADE OF SERVICE TO THE COMPUTER USER!

486SX-20 SYSTEMS - $1795.00!

Now You can enter the world of 486 computing at a reasonable cost!



the OWL SUPER ATOM - 486

High Powered Computing from a local, well established company.

- 33MHz / 50MHz i486 based Systems with Socket for Weitek CoProcessor
- System and Video BIOS in Cache
- Large Tower Case : (33MHz, FCC Class B) - (50MHz, FCC Class A)
- 230 Watt Power Supply & 8 Option Slots
- System Price Includes: 40MB HD, 4MB RAM, Std. Resolution Color VGA Monitor, High Resolution VGA Card, 2 High Density FD's, MS DOS 5.0

$1795 / $2095 / $2695 / $3595

486SX-20 ISA 486DX-33 ISA 486DX-50 ISA 486DX-50 EISA

- 105MB HD Upgrade Add $160.00
- Super VGA Upgrade Add $95.00

3- YEAR WARRANTY on all systems! Including One Full Year on Parts and Labor. Manufactures 3-Year Warranty on All Hard Drives

OWL SUPER ATOM - 386

- 25/40MHz 386DX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 4MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

$1565/$1645

25MHz 40MHz

OWL SUPER ATOM - SX

- 16/25MHz 386SX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 2MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

$1295/$1349

16MHz 25MHz

386-SX Notebook Computers

20MHz, 60MB HD, 1.4MB FD, 2MB RAM(exp. to 5MB), VGA 640X480 LCD w/32 shades of gray. Ports: 2 Ser, 1 Par, 1 VGA, DOS & Windows, 7.7LBS!

$1545

386-DX Notebook Computers

33MHz, 120MB HD, 1.44MB FD, 32KB CACHE, 4MB RAM(exp. to 16MB), Std. VGA LCD w/32 gray, Ext. Keypad inc., DOS & Windows, 7.7LBS!

$1995

OWL COMPUTER SERVICES

5950 Keystone Drive
Bath, PA (215)-837-1917

Kids & Us - RadioShack®	*St. Onge Systems*	*Computers & Games*
Pottstown Ave., RT. 663	*Wescosville*	*Muhlenberg Shopping Plaza*
Pennsburg (215)-679-3389	*Call for Appt. (215)-481-9775*	*Reading (215)-929-0540*

DISK DRIVES

Bonus! Special Bundled Software with any Disk Drive Purchase!

Floppy Drive Systems

The Highest Quality for Years of Service

Drive 0 Systems (Half Height, Double Sided,

SOLD OUT!
WE NEED CONTROLLERS! ~~$180.~~

IF YOU HAVE 502 CONTROLLERS, CALL US!

Drive 1 Systems (Half Height, Double Sided, Direct Drives) **$115.**

New 3.5", 720K Drives for OS-9 with case & Power Supply **$129. SALE!**

Drive 1 Systems have drive, case, power supply. (You may require optional cable and/or DOS chip to use)

Special for 0/1 Combos (0,1,2,3) $199.

(WITHOUT CONTROLLER)

HALF- HEIGHT DRIVE UPGRADES FOR RS HORIZONTAL CASES

Why only double the capacity of your system when you can triple in the same case? Kit includes: double-sided to fit your case, chip to run both sides of new drive, hardware, and detailed instructions. Easy! Takes only 5 minutes!

Model Only $119.

500 , 501, or 502

All drives are new and fully assembled. We ship only FULLY TESTED and CERTIFIED at these low prices. We use Fuji, YE Data, and other fine brands. No drives are used or surplus unless otherwise stated to you when you order. We appear to be the one of the few advertisers in Rainbow who can truly make this claim. We have 7 years experience in the CoCo disk drive market! We are able to provide support when you have a problem.

Drives 1 Year Warranty

OWL Phones

Order Numbers (only)
1-800-245-6228
1-215-682-6855
Fax: 1-215-837-1942
Technical Help
1-215-837-1917

OWL WARE Software Bundle

Disk Tutorial/Utilities/Games

DISK TUTOR Ver 1.1

Learn how to use your disk drive from this multi-lesson, machine language program. This tutor takes you through your lessons and corrects your mistakes for a quick, painless disk drive introduction. (This professionally written tutor is easily worth the bundle's total price.)

3 UTILITIES

A copy verify, copy, and DOS utility.

2 GAMES

We will select 2 games from our stock. These are sold for more than $20 each.

Do not mistake this software with cheap "Public Domain" software which others offer. All of this software is copyrighted and professional in quality. The tutor is unique with us and has helped thousands of new users learn their disk drive.

only $27.95
(or even better)
only $6.95 with
any Disk Drive Purchase!!

512K Upgrade

Again at a popular price. Fully assembled and tested before shipping. Easy to install. Uses fast 120 ns. chips.

SALE $79.

Now includes **memory test, Ram Disk Lighting, Printer Lighting, and Back-up Lighting.** All with an upgraded manual exclusive with OWL!

Our prices include a discount for cash but do not include shipping.

OWL-WARE has a liberal warranty policy. During the warranty period, all defective items will be repaired or replaced at our option at no cost to the buyer except for shipping costs. Call our tech number for return. Return of non-defective or unauthorized returns are subject to a service charge.

OWL-WARE
P.O. BOX 116
Mertztown, PA 19539

COCO CONSULTS

MARTY GOODMAN

CoCo 4-pin DIN	Buffer DB-25	Description
1	n/c	(not connected)
2	20	(busy signal handshake from buffer to CoCo)
3	7	(ground)
4	3	(serial data to buffer)

Figure 1: Printer Cable

Where to Find 6309 Info

Where can I find more information about the Hitachi 6309? What about support for using it under Disk BASIC?

Dennis McMillian (COCOKIWI)
Pittsburgh, California

A Chet Simpson (HYPERTECH) of Henderson, Nevada, posted on Delphi some new material on the technical aspects of the 6309 and on how to use its features with Disk BASIC. In both the CoCo and OS-9 SIGs can be found versions of his 35-page document that details the Hitachi 6309 opcodes. Chet's work is extensive and very professional. It is a combination of information obtained from the original document from Japan that we've had for the last couple of months plus information he obtained by experimenting with the 6309 over the last month. Chet also provides programs to patch the NMI routine of Disk BASIC in order to allow the 6309 to operate in native mode. Chet's package is a total of about 130K characters.

The entire package is available in the CoCo SIG in TC archive format (about 90K). The text file (minus the BASIC patch programs) is available in the CoCo SIG in an uncompressed ASCII file (about 130K), and in a `.ZIP` file (about 25K) for the convenience of users of other computers. The entire package is also available in the OS-9 SIG in an AR archive file (about 45K). Chet says he is preparing other material relevant to the 6309, including patches for various BASIC programs. Specifically he is working on patches to some graphic display programs that will take advantage of the vastly faster block-move capabilities of the 6309.

Again a reminder for those interested in the 6309: Burke & Burke has a set of patches for OS-9 bundled with a 63B09E chip with installation instructions called PowerBoost. Burke & Burke also has available a book on the 6309. Cer-Comp, at last report, is putting the finishing touches on a Disk BASIC assembler for the 6309.

6309 Information

Many of you have asked what is involved in using the 6309 with Disk BASIC, and whether or not it would involve too many patches to specific software. Art Flexser (ARTFLEXSER) has provided the following information:

With patches for the NMI routine of Disk BASIC (to take account for the extra register pushed on the stack when an NMI is enconntered) Disk I/O will work. You have to patch some constants in the sound routine to get proper tones and also patch the baud constants in the "bit banger" RS-232 generator of BASIC to get your printer port to work at a proper speed. After this, all ordinary programs written in BASIC (apart from some with hidden machine-language code that has critical timing loops) should run fine and benefit from the 5 to 10 percent overall speed increase available due to the more efficient cycle times of the 6309 in native mode. Assembly-language programs such as terminal programs and word processors require patches to their baud constants for proper operation of a printer and modem if the modem is being used through the 4-pin "bit banger" port on the CoCo. These programs also need to have the Disk I/O routines patched to reflect the extra register pushed on the stack when an NMI is called. Typically `LEAS 10,X` has to be changed to `LEAS 12,S`, either in Disk BASIC's NMI routine, or in the NMI routine that the specific program might set up on its own.

By far the biggest benefit of the 6309 is in cases of programs specifically rewritten to take best advantage of it: i.e. graphic display programs that do a great deal of block moves of data. Certain highly specialized programs that do mathematical computations might also benefit from the 32-bit divide in the 6309, though they'd have to be extensively rewritten to take advantage of that capability.

Printer Buffer Needs Help

I just got a Quadram Microfazer 16K serial-to-parallel buffer. It has inside one 4164 chip and seven empty sockets. It refuses to work when used with my modem cable to hook it to my CoCo's serial port. How can I make it work? Can it be upgraded to have more memory?

I recently bought an old Color Computer 2 at a swap meet and found inside it a little board plugged into the VDG socket that had on it a VDG and a 24-pin EPROM chip, along with several smaller chips. Is this an 80-column board?

Wayne Thompson (WTHOMPSON)
Sachse, Texas

A Modem cables and printer cables for use with the 4-pin bit banger port are wired differently, and they cannot be exchanged. Try a cable from the 4-pin DIN socket of your CoCo to the DB-25 serial connector of the buffer that is wired as shown in Figure 1.

You can upgrade that buffer to a full 64K capacity merely by plugging in seven more 4164 chips of virtually any speed (that buffer can even use 200ns speed chips).

The board yon describe is not an 80-column board. Rather, it is a lowercase board that allows the VDG of a CoCo 1 or 2 to access extra character data in the EPROM chip on it to put up true lowercase characters on the screen. Such boards usually had external switches that would turn on or off the true lowercase function and also allow inverting the video for light on black display. The boards usually were specific for a given model of CoCo due to mechanical considerations.

Double-Sided Access

How can I gain access to both sides of a double-sided drive with Disk BASIC? I am using a Color Computer 3.

Robert Schmidt (ROBERT84)
Grand Forks, North Dakota

A There are two approaches. One is quick and dirty and somewhat limited in its capabilities, but without cost. The other is elegant, convenient and compatible with a wide variety of applications, but costs nearly $50. If you are running Disk BASIC 2.0 nn your CoCo 3, at the BASIC prompt type the following:

```
POKE &HD7AC,&H41
POKE &HD7AD,&H42
```

If you have Disk BASIC 2.1, type:

```
POKE &HD89F,&H41
POKE &HD8A0,&H42
```

At this point Drive 2 will be the back side of physical Drive 0, and Drive 3 will be the back side of physical Drive 1 (the preferred configuration for Disk BASIC). You will be able to load and save BASIC programs from the back sides of disks in those drives (provided you do in fact have double-sided drives). The limitations of this quick fix are that you still will be using only 35 of the 40 tracks available; the fix will not work with many applications written in assembly language; and occasionally during a disk backup or copy from one side of the same disk to another, you might crash the destination disk.

The elegant and professional approach is to buy a copy of *ADOS 3* or *ADOS 3 Extended* from SpectroSystems (inquire with ARTFLEXSER on Delphi). This will fix your system for 40-track and double-sided operation in a way that is compatible with most (though not all) applications with Disk BASIC, eliminate the problem of occasional crashed copies, and add a host of extremely convenient features to Disk BASIC. *ADOS 3 Extended* adds RAM disk capability to Disk BASIC if you have 512K. When you include the cost of having an EPROM burned with *ADOS 3 Extended*, the total cost comes to about $50 — but just about everyone who has it says it's well worth it.

Which BASIC is Which?

How can I upgrade an FD-500 disk controller from Disk BASIC 2.0 to Disk BASIC 2.1?

Mike Nelson (MICHAELJN)
Lancaster, Pennsylvania

A The version of BASIC you get depends on the ROM installed in the disk controller and whether you are using a CoCo 1 or 2 or a CoCo 3. CoCo 1's and 2's come up with Disk Extended Color BASIC Version 1.0 or 1.1, depending on what ROM is in the disk controller. CoCo 3's, with the same ROMs in the disk controller, come np with versions 2.0 or 2.1. To change from version x.0 to version x.1, you must buy a licensed ROM chip with the new version in it. CoCoPRO! may sell these ROM chips, which come in 24- aud 28-pin varieties. If you have a Tandy controller other than the FD-502, you need the 24-pin variety of Disk BASIC 1.1. Most third-party controllers take 28-pin ROM chips, though some also accept 24-pin ROM chips and a few very early models of third-party controllers actually require a 24-pin ROM chip. Your FD-500 controller uses a 24-pin ROM chip.

Note: When looking for the ROM chip on a controller, do not confuse it with the floppy disk controller chip. Newer controllers use a 28-pin WD1773 disk controller chip. This is not the ROM chip! There is not a lot of difference between Disk BASIC 1.0 and 1.1. A few minor bugs have been fixed, and one single feature (the `DOS` command used to boot OS-9) has been added. *ADOS* and *ADOS 3*, by the way, are fundamentally patched versions of Disk BASIC 1.0, with bags cleaned ap and a custom `DOS` command added.

68B09E Sources

Where can I buy a 68B09E microprocessor chip?

Rick McNabb (RICKMAC)
Citrus Heights, California

A BG Micro (214-271-5546) used to carry the 68B09E for about $4 per chip. If you are replacing your 68B09E, you might waut to consider getting a 63B09E to benefit from its lower-power operation and to be able to play with the new commands it offers.

JDR Microdevices listed the 68B09E in the last catalog. It was priced at $6.95. JDR does sell items in quantity one via mail order, though you may be charged a moderate handling fee.

Orchestra-90CC Files

I have a PC compatible with a 1.2MB drive. I wanted it to format disks that your `MS2COCO` *file-transfer program could read. I found that if I formatted disks in that drive with the syntax:*

```
FORMAT A: /4 /1
```

(specifying low density with the /4 switch and singled sided with the /1 switch), I could produce a disk that, after I transferred files to it from my hard drive, I could read with your program. I just thought you'd like to know.

Now for my question. I downloaded a number of Orchestra 90 *files from the CoCo SIG using my PC compatible, then transferred them to my CoCo using Marty Goodman's file-conversion utilities. I then used Mike Ward's* `OCNVRT` *to convert the ASCII files into binary* Orchestra 90 *files. However, they still won't play with* Orchestra 90. *I get the dreaded ERR 7 halt that says the file is not found on my disk. Any ideas?*

Randy Schmidt (IBMUSER)
Canton, Ohio

A *Orchestra 90* requires the music files it uses not have an extension. So if you have on your disk a music file named `SONG.EXT`, rename that file to `SONG` and it should work fine.

Upgrade the MPI?

I'm about to buy a used Multi-Pak Interface from a fellow who says it is not upgraded but that it works well with a CoCo 3. What is the upgrade for, and do I really need it?

Tony Walls (TONYWALLS)
Janesville, Wisconsin

A This is a frequently asked question. In several RAINBOW articles I have given the details of what is involved in upgrading both models of Multi-Pak Interface. The older, bigger interface (Radio Shack Cat. No. 26-3024) that comes in both gray and white case varieties is easily upgraded by replacing a single socketted PAL chip. This chip is available from CoCoPRO! The newer, smaller Multi-Pak Interface (Cat. No. 26-3124) is more complicated to upgrade, requiring a trace cut and the addition of a patch board with several wires to be soldered. CoCo PRO! sells kits for this upgrade (the schematics are available on the Delphi CoCo SIG). My advice is that if you are using a Multi-Pak Interface with a CoCo 3, you should upgrade it regardless of whether or not it appears to work OK. If you fail to upgrade, your system will experience bus coufflicts with the GIME chip, which some say can harm the GIME chip.

Martin H. Goodman, M.D., a physician trained in anesthesiology, is a longtime electronics tinkerer and outspoken commentator — sort of the Howard Cosell of the CoCo world. On Delphi, Marty is the SIGop of THE RAINBOW's CoCo SIG. His non-computer passions include running, mountaineering and outdoor photography. Marty lives in San Pablo, California.

Most CoCo users are continually looking for ways to improve their systems. And the hardware design of the Color Computer makes it easy to add improvements. Projects may range from simple power indicators to home control systems. If you have built a circuit or devised a modification for the Color Computer, why not share it with others?

We are now making tentative plans for the March 1993 issue of THE RAINBOW and are accepting submissions appropriate for that issue's theme, Hardware. We are also interested in articles discussing current hardware trends *as applicable to the CoCo*. All submissions must be received by us no later than November 27, 1992, and must follow our standard submission guidelines (see Page 3 for details and address). Be sure to include schematics, parts lists, and all figures and tables.

We'd also like to see any other programs you have written (submitted material must be the original work of the submitting party, or submitted with written permission). All submissions are evaluated and considered for publication in future issues.

The following products have recently been received by THE RAINBOW, examined by our staff and issued the Rainbow Seal of Certification, your assurance that we have seen the product and have ascertained that it is what it purports to be.

CF83-12: RS-DOS File Handling, allows the creation of programs in Forth that can be executed using LOADM and EXEC without requiring the user to first load CF83 Forth. No license required for distribution of the resulting programs. Requires *CF83 Forth. BDS Software, P.O. Box 485, Glenview, IL 60025-0485, (708) 998-1656; $13 with printed manual, $10 with manual on disk; Canadian orders add $3, all other foreign orders add $10; all funds U.S.*

CF83 Public domain Disk No. 1, a group of programs for the *CF83 Forth* system. Included are programs for converting *CF83 Forth* blocks to Disk BASIC (and vice-versa), text characters for PMODE graphics, a BASIC utility for dumping disk directories to ASCII files, benchmark comparison software for BASIC, assembly language and *CF83 Forth*, text-screen control, and a *CF83 Forth* demo. Requires *CF83 Forth. BDS Software, P.O. Box 485, Glenview, IL 60025-0485, (708) 998-1656; $3.*

CoCo Cassette #119, a variety of programs for the CoCo 1, 2 and 3. This issue includes *Titles*, for adding title screens to BASIC programs; *Picture Maker*, for drawing PMODE4 pictures; *Square*, a logic game; *Disk Checker*, checks for bad sectors; *Hints*, have a conversation with your computer (requires S/SC); *Lotto Predictions*; *Pocketball*, an arcade-style game; *RAMTest*, a memory checker; *CoCo Directory Manager*, disk utilities; and *Mutant Attack*, a machine-language arcade space game. *T & D Subscription Software, 2490 Miles Standish Drive, Holland, MI 49424, (616) 399-9468; $8.*

CoCo Friends Disk Magazine, a magazine on disk for Disk BASIC users. Published monthly, the disk includes feature articles, puzzles, programs, reviews and more. *CDFM* requires a CoCo 3, an RGB monitor and a disk drive. *Rick's Computer Enterprise, P.O. Box 276, Liberty, KY 42539, (606) 787-5783; $6 per issue, $16 for three issues and $30 for six issues.*

The Rainbow Seal of Certification is open to all manufacturers of products applicable to the Tandy Color Computer, regardless of whether or not those companies advertise in THE RAINBOW. By awarding the Seal, we certify the product exists—we have a sample copy and have examined it. However, this does not constitute any guarantee of satisfaction. As soon as possible, these products will be forwarded to reviewers for evaluation.

The CoCo 3 supports two approaches to controlling the computer's action when the BREAK key is pressed. When writing programs, you can use the CoCo 3's ON BRK statement, or you can disable the BREAK key altogether. To disable the CoCo 3's BREAK key, enter

```
POKE &HE414,0:POKE &HE42A,0
```

To enable the BREAK, perhaps after critical processing is complete, enter

```
POKE &HE414,3:POKE &HE42A,3
```

Feature Article

THE RAINBOW interviews tony podraza

Effective communication is the cornerstone of any Community, and the Color Computer market is certainly no exception. In support of this and to get the word out about groups and clubs, we asked Tony Podraza, president of the Glenside Color Computer Club, if he would agree to an interview to be published in THE RAINBOW. Tony did not hesitate in saying "Yes!"

The interview took place via fax machines (another marvel of rapid communication) over a period of two weeks. In some cases the answers Tony provides are the result of discussions with other Glenside members. We believe you'll be as interested in these representative views as we are.

THE RAINBOW: *We understand the Glenside Color Computer Club is one of the largest users groups in the CoCo Community. Just how many members do you have?*

Tony: Currently there are 146 club members who live in the United States. Of these, about 30 are in this general area and regularly attend meetings. The rest are out-of-state members and live in places from California to New Jersey. And we don't restrict membership to U.S. residents; Glenside has four members who live in the Netherlands and Canada.

THE RAINBOW: *Wow! Perhaps Glenside is the largest users group for CoCo owners. What makes the club so strong?*

Tony: Glenside's strength is just that — Glenside. That sounds a bit like the hound chasing its tail, doesn't it? Let me explain. The total of Glenside is the sum of its parts: the members make the difference. The support we give each other, the information we pass to one another via the newsletter, the way we work to help other CoCo users all add up. As one member pointed out, "the support that Glenside is ready to give to its members and the Community has continued to strengthen it from its founding."

Glenside's exposure to the Community has helped to strengthen it as well. As you know, the club has hosted at least four RAINBOWfests and was officially represented at the second Atlanta CoCoFEST and the recent Chicago CoCoFEST. All this highlights the commitment of the club to the support of the Community.

THE RAINBOW: *Speaking of support, how are membership dues applied and what can club members expect from Glenside?*

Tony: Our annual dues are used to cover a number of expenses, the first and foremost of which is *CoCo~123*, the newsletter that keeps Glenside in touch with Glenside. Second, these funds help us keep a meeting place available. We meet at the Glenside Public Library, from which we take our club name. Another expense is the maintenance of our club BBS. Also as a benefit of membership, we distribute a "member's utility disk" (written by Joel Hegberg just for Glenside), and we have plans to distribute three more disks to the membership at no extra charge. Finally, but certainly not least, we use these funds for whatever else we can find to support the club and the Community. An example is our involvement in CoCo shows. It is to this support that the treasury is directed. Well . . . there is the one exception. We do have an annual social event, most recently a picnic, for which the club provides meat and rolls — a whopping $25-$30 at best guess.

Even if they don't live in our area, new members can take advantage of most of these things for the membership dues they pay. *CoCo~123* has received a number of kudos from people in the publishing world and has become a strong medium of club interaction for the "members at large." Indeed, it was an article written by a local member that brought one gentleman *back* to our exhibit in Atlanta in 1991 after he had hemmed and hawed about joining. As he related to me, he had already "found about $15 worth of information in the newsletter" that he had received as part of a free handout package.

We also have the "Glenside Club Plan" that invites vendors to submit ad copy for publication in the newsletter. Reciprocal token discounts on products are welcome for this service but are not required. Member savings from this plan are an added benefit.

Other methods of interaction with the Community include the SIGs on Delphi, where a number of members gather; networks set up through STGNet; and hopefully, in the future, FIDO.

It is important to note that membership is not only a privilege but carries responsibility as well. We believe every member has something to share, and we encourage participation via the newsletter and club meetings, if possible.

THE RAINBOW: *Describe a typical meeting.*

Tony: Meetings open with the sounding of the electronic presidential gavel — actually a three-sound siren repackaged in a gavel-shaped housing sporting a Tandy RAM-button handle. After that bit of nonsense is out of the way, visitors and new members are recognized, then we hold a short business meeting. After this we share news and views, questions and answers, and maybe a little gossip is supported (but *never* started). During the last half of the meeting, a scheduled demo takes place. This could be a run-through of a program or a hardware demo.

After the meeting, the majority of attendees adjourn to the local restaurant for "milk and cookies." This is where the brainstorming starts to take place. It was at one of these sessions that the RGB-A-to-TTL video adapter was spawned.

THE RAINBOW: *Earlier you mentioned a club BBS. Would you give our readers some of the details?*

Tony: Yes. The Glenside CoCoRama BBS has been in operation since 1986, and its sysop is Dave Barnes. The system is online 24 hours every day and can be reached at (708) 587-9837. Inbound callers should sets their terminals at 8-N-1 for 300-, 1200- or 2400-bps access. The board can be used by anyone, though file downloading on the first access is limited.

In addition to CoCoRama, there are at least two other BBSs currently operating in direct support of the club. They are listed in the newsletter and also on the CoCoRama logoff screen.

THE RAINBOW: *We have seen some Color Computer users groups that are really nothing more than high-seas boarding parties. What is Glenside's position on the subject of software piracy?*

Tony: I'm glad you asked. As responsible people, we need always to be aware of our actions and their effect on others. The wanton passing around of commercial software does nothing but discourage authors and vendors from continuing to support the market. Indeed, where is the market when you can get the product from your buddy across town? When authors no longer produce new software because there is nothing to be gained from it, who's going to write the good stuff? Your buddy?

For the record, Glenside doesn't support software piracy in any form. As stated in our bylaws, anyone fonnd using the club system for such practice will he disassociated from the clnb. These [now] non-members forfeit their annual dues, do not receive any club benefits or services, and may not rejoin. We don't have a problem.

THE RAINBOW: *What is the personal "toll" on you as president of Glenside, and what advice would you offer those who seek to start their own clubs?*

Tony: Time and money: I spend many hours on the phone asking and answering questions, and arranging demos and the like for the meetings. The real winner here is the phone company — that's where the money part comes in. Of course for every hour I spend on Glenside, that's another hour not spent with my family. This can be disastrous if not balanced carefully. Finally, there is the toll that ethics, politics and tactfulness puts on a person. I have to be very careful in what I say and do so as to not misrepresent information. CoCo users are a close-knit family, and I have no intention of knocking the feet out from under anybody. The efforts put forth to maintain an organization like Glenside without rocking the boat may not be monumental, but they are very real.

Advice for others? First, don't wait for someone else to start it! If you know one other CoCo user in your area and there are no organized users groups, get together with that person; you now have a "club." Find a place to meet — just about anyplace will do. Then get the word out; put notices in the library, the grocery store . . . anywhere you can find to let others know you are there. Develop a meeting format that the majority of members is comfortable with — it doesn't have to be formal but should be interesting. Also develop a set of realistic goals and guidelines. If you want the club to be ongoing, the core of the club needs to be dedicated and energetic.

Operating a users group may also require some money to maintain things such as a roof over your heads, mailing newsletters, and purchases for the benefit of the club. So after you have an idea what your expenses will be, be prepared to initiate an annual amount for membership dues. Above all, try to strike a balance. Remember that the club does not belong to any one member but is the medium through which all members benefit in terms of knowledge and understanding of the CoCo, regardless of their individual preferences.

THE RAINBOW: *What percentage of Glenside's membership uses OS-9?*

Tony: Judging from our meetings and responses from the membership, I would say 75 to 85 percent of Glenside's members use OS-9 for 85 to 90 percent of their work with the CoCo. I would estimate that less than 25 percent of the club membership uses Disk BASIC exclusively. This is not to say that the rest of the Community follows suit — there are a lot of Disk BASIC users out there. But as knowledge of the CoCo's capabilities under OS-9 grows, the percentages are starting to lean in that direction.

THE RAINBOW: *Debate between the diehard OS-9 and Disk BASIC camps is often quite intense. Would you say this schism is an inhibiting factor in any potential growth of this market? If so, what possible solutions would you suggest?*

Tony: No, not really; not any more than the GUI environment versus DOS environment is an inhibiting factor in the IBM (and compatible) world. While it appears to be true that the future growth of this market lies with OS-9, it is very beneficial for users to know a little about Disk BASIC before attempting to use OS-9. What is important for people to understand is that OS-9 is not an enemy but merely another environment in which to use the hardware. Far too many Disk BASIC users end up disliking OS-9 before they receive the support necessary for a clear understanding of the system. One of Glenside's goals is to provide that support for both Disk BASIC users and those who choose to venture into the world of OS-9.

THE RAINBOW: *It is generally accepted that MS-DOS is the most widely used platform across the computing industry. For whom would you say the Color Computer is a viable machine and why?*

Tony: For the same people for whom it has been a viable machine since its inception — those who are on a tight budget and want to learn about computing; hobbyists who want to experiment; those who want to find out what true multitasking/multiuser capabilities are like without having to buy a 386 or mini-mainframe. Of course that's the serious side. The CoCo is also a pretty fine game machine (especially according to Tandy). Unfortunately this aspect has also been the area of greatest advertising for the machine — to the extent that the real power of the 6809 has been hidden from the general public.

THE RAINBOW: *Yes, it is sometimes difficult to enjoy playing games knowing they have been somewhat of a stigma to growth. How has Tandy's discontinuation of the Color Computer affected Glenside?*

Tony: It has increased the need for our existence and has made us a stronger voice in the community, I tbink, because of our willingness to *be there* for the Community. It goes without saying that the product knowledge and support from the local retail outlets for the Color Computer has never been what it is for the other systems. Now the average salesperson's desire to support a non-income producing item is in the twilight zone. When you need someone to answer a question or solve a problem, to quote a phrase, "who ya gonna call?" It ain't Ghostbusters! It'll be your local club co-member. And Glenside is just *one* of those local clubs. We've simply got a bigger "ear" to hear the callers.

THE RAINBOW: *What can the average person do to ensure continued enjoyment of his CoCo?*

Tony: Link up with other CoCo users; be imaginative in using the CoCo. Robert Kennedy once said,

Mike Knudsen (second from left), author of UltiMusE, *regularly attends the Glenside Color Computer Club meetings.*

"Some people see things and ask 'why?' I dream things that never were and ask 'why not?'" Tell others what you are doing with your CoCo and share your discoveries. You may have found something the next person has been working for a long time to solve. The joy of discovery cannot be replaced.

Communication is the key to the survival of the CoCo. Get in touch with other clubs and users through whatever means you have, whether that is by telecommunications, newsletters, subscriptions to dedicated publications or road trips to gatherings. There are many "orphaned-computer" users groups out there, many of which are thriving simply because they continue to communicate.

THE RAINBOW: *We would like to know what comments Glenside has about THE RAINBOW.*

Tony: For starters, how about helping to keep the hardware in the active community by opening a "classifieds" section? I belong to a number of national organizations that produce major publications, and all of them publish such a section.

The current format is really colorful, but the paper it is printed on is not easily preservable or long-lasting. Possibly you could do something there. And we'd like to see a return to the 8½-by-11 size, perhaps with stitch binding.

We like the assembly-language programs and would like to see them continue for both Disk BASIC and OS-9. The inclusion of C-language programming articles would be nice. I particularly enjoy hardware articles, so those should stay around, too.

Despite the publisher's comments a few months back regarding the 68xxx machines, I believe items relating to them should be considered for publication. The people who have made the CoCo what it is are doing the same for those computers. To shy away from 68000-based computers because "bunches and bunches of software" are not currently available is not valid. And because of the commonality of OSK with OS-9, I feel that the software for these machines should be covered in the venue of the publication from time to time.

Finally, there is a lot of murmuring out here about the possibility of a THE RAINBOW becoming a merged CoCo/MS-DOS publication, even though most readers are just happy to be able to get the magazine. Consider going to a bi-monthly schedule instead, and consider reviewing and reducing prices on products. Of course, I don't know the logistics and costs involved; I'm just the elected messenger.

Thank you, Tony, for taking the time to answer our questions and for giving us some honest feedback. Your answers are important to us, and they should be of interest to our many readers as well. Yes, there are a lot of things we must consider with every decision we make, and while these decisions make sense to us, not everyone agrees with them. This doesn't mean we won't make changes, but we must be careful when we do. Regarding a merged publication: we believe those fears will prove unwarranted. THE RAINBOW is a CoCo publication.

Reaching Out

Now is as good a time as any to make sure you stay in contact with other users of the Color Computer. The currently available means to this end include joining users groups, writing letters, subscribing to publications, and taking advantage of online services such as Delphi, CompuServe and CoCo BBSs. For more information about CoCo clubs and BBSs that may be in your area, see "Intercom" elsewhere in this issue of THE RAINBOW.

Remember that many clubs accept memberships from distant users. Those interested in joining Glenside can write to Glenside Color Computer Club at 119 Adobe Circle, Carpentersville, IL 60110-1101. Be sure to include $12 to cover the annual membership dues.

Chris Hawks (center) of Hawksoft answers another user's question.

BUGGY *FROM COVER*

within the screen boundaries. Anything greater than 191 wraps around to zero and any value below zero wraps to 191. (Can you figure out why we don't have to check the *x* coordinate?)

If the current bug's new *y* coordinate is the same as the "zapper" location, the program goes to ZAP where that point is PRESET. Now we have to move all the rest of the current locations in LOC down by one to replace the bug just lost. Since Register Y points to the next bug's location, we reduce it by two to reflect the current bug; now load the coordinates for the next bug and save them at the current location. Keep doing this for the rest of the array, then decrease the number of bugs by one.

Finally, we have to shift each bug's direction toward the current bug's location (still in $BE and $C0). The distance from the current bug to each bug is compared in the *x* and *y* directions. If the current location is less than a bug's *x* coordinate, the uew *x* change for that bug is -1 to move it toward the current location; if the current location is greater, the change is +1; and if the *x* distance is the same, the change is 0. The *y* coordinate changes are computed in the same manner, and both changes are stored in the DEL array. After the ZAP routine is completed, the program goes back to LOOP1. If no bugs were on the zapper, COUNT1 (the nnmber of times) is decreased. If no bugs were on the zapper during any of the Number Of Times, the program goes to the CHANGE routine to compute new location changes of +1 or -1.

After you've entered the program in Listing 2, check it for errors by entering A/NO/NS/WE. Then save it using W BUGS.SRC and assemble it using A BUGS.BIN /NS/WE.

The BASIC program shown in Listing 1 is the driver for the machine-language program. It loads the assembled program (if necessary) and executes it. The variables you can change are:

N — number of bugs (1-250)
L — location of the zapper (0-191)
T — number of times in the same direction (1-255)

If you're just itchiu' to increase the number of bugs beyond 250, you'll have to make NUMB two bytes in length and increase the LOC and DEL arrays accordingly.

For experimentation, try changing the zapper location: make it one point instead of a line, and revolve that point as if it were on a circle. How does the swarm move then?

If you have any suggestions for future articles, or questions about any article, don't hesitate to write.

Bill Nee bucked the snowbird trend by retiring to Wisconsin from a banking career in Florida. The success of his 13-part series, "Machine Language Made BASIC" (July 1988 to July 1989), prompted him to continue writing articles about Color Computer machine-language programming. You may contact Bill at Route 2, Box 216C, Mason, WI 54856-9302, (715) 746-2952. Please include an SASE when requesting a reply.

64K Disk

Listing 1: BUGDRIVE

```
1 'BUGS DRIVER
2 'BY WILLIAM P. NEE
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 CLEAR200,&H6000-1
20 IF PEEK(&H6004)<>16 THEN LOAD
M"BUGS":POKE &HFF40,0
30 X=RND(-TIMER)
40 N=200:'NUMBER OF MOTHS
50 POKE &H6000,N
60 L=96:'LOCATION OF ZAPPER
70 POKE &H6001,L
80 T=5:'NUMBER OF TIMES
90 POKE &H60AB,T
100 PMODE4,1:PCLS:SCREEN1,1
110 EXEC &H6004:CLS
120 FOR X=1 TO 100:NEXT X:CLS
130 PRINT@200,"THAT'S ALL FOLKS"
140 GOTO 140
```

Listing 2: BUGS

```
00100 ORG     $6000
00110 RANDOM  MACRO
00120         PSHS    Y
00130         LDD     $116    RANDOM SEED
00140         LDX     $118    RANDOM SEED
00150         JSR     $9FB5   D*X->Y,U
00160         LDD     $BF74   RANDOM CONSTANT
00170         LEAY    D,Y     ADD IT TO Y
00180         STY     $116    RESAVE IT
00190         LDD     $BF76   RANDOM CONSTANT
00200         LEAU    D,U     ADD IT TO U
00210         STU     $118    RESAVE IT
00220         LDB     $117
00230         ANDB    #1
00240         INCB
00250         LSLB
00260         SUBB    #3      -1,+1
00270         PULS    Y
00280         ENOM
00290
00300 PSET    EQU     $9374
00310 PPOINT  EQU     $933C
00320 RND0    EQU     $BF3B
00330 NUMB    FCB     200
00340 LEN     FCB     96
00350 COUNT1  RMB     1
```

```
00360 COUNT2  RMB     1
00370
00380 START   LDY     #LOC    LOCATION ARRAY
00390         LDB     #$FF
00400         STB     $B5     CURRENT COLOR
00410         LDB     NUMB
00420         STB     COUNT1
00430 RANLOC  LDD     #255
00440         JSR     $B4F4   MOVE TO FP1
00450         JSR     $BF1F   RANDOM(255)
00460         JSR     $B3ED   MOVE BACK TO D
00470         STB     $BE      AND SAVE IT
00480         LDD     #191
00490         JSR     $B4F4
00500         JSR     $BF1F
00510         JSR     $B3ED
00520         STB     $C0
00530         JSR     PPOINT  CHECK THAT LOCATION
00540         JSR     $B3ED
00550         CMPB    #5      ALREADY PSET?
00560         BEQ     RANLOC  TRY AGAIN
00570         LDA     $BF
00580         LDB     $C0
00590         STD     ,Y++    STORE THE COORDINATES
00600         JSR     PSET     AND PSET THE POINT
00610         DEC     COUNT1
00620         BNE     RANLOC
00630
00640 CHANGE  JSR     RND0    KEEP THINGS HONEST
00650         LDB     NUMB
00660         STB     COUNT1
00670         LDY     #DEL
00680 DELX    RANDOM
00690         STB     ,Y+     SASVE RANDOM DX
00700 DELY    RANDOM
00710         STB     ,Y+     SAVE RANDOM DY
00720         DEC     COUNT1
00730         BNE     DELX
00740
00750         LDB     #5      NUMBER OF TIMES
00760         STB     COUNT1
00770 LOOP1   LDY     #LOC
00780         LDB     NUMB
00790         STB     COUNT2
00800 LOOP2   LDD     ,Y      CURRENT X,Y LOCATION
00810         STA     $BE
00820         STB     $C0
00830         CLR     $B5     NO COLOR
00840         JSR     PSET
00850 GETX    LDA     $BE     CURRENT X COORDINATE
00860         ADDA    500,Y   CURRENT DX
00870         STA     ,Y+     NEW X COORDINATE
00880         STA     $BE
00890 GETY    LDB     $C0     CURRENT Y COORDINATE
00900         ADDB    500,Y   CURRENT DY
00910         CMPB    #192    KEEP IT
00920         BEQ     ZEROY
00930         CMPB    #-1      WITHIN THE SCREEN
00940         BEQ     MAXY
00950         BRA     GETY2
00960 ZEROY   CLRB
00970         BRA     GETY2
00980 MAXY    LDB     #191
00990 GETY2   STB     ,Y+     NEW Y COORDINATE
01000         STB     $C0
01010         LDB     #$FF
01020         STB     $B5
01030         JSR     PSET    PSET NEW COORDINATES
01040         LDB     $C0
01050         CMPB    LEN     ON THE ZAPPER?
01060         BEQ     ZAP
01070         DEC     COUNT2
01080         BNE     LOOP2
01090         DEC     COUNT1
01100         BNE     LOOP1
01110         LBRA    CHANGE
01120
01130 ZAP     CLR     $B5
01140         JSR     PSET    CLEAR THE POINT
01150         LEAY    -2,Y    BACK TO CURRENT COORDINATES
01160 ZAP1    LDD     2,Y     NEXT SET OF COORDINATES
01170         STD     ,Y++    SAVE THEM AS CURRENT ONES
01180         DEC     COUNT2
01190         BNE     ZAP1
01200         DEC     NUMB    ONE LESS BUG
01210         BEQ     DONE
01220
01230 NDEL    LDY     #LOC
01240         LDB     NUMB
01250         STB     COUNT2
01260 NEWDX   LDA     $BE     CURRENT X COORDINATE
01270         CMPA    ,Y      COMPARE TO THE ARRAY
01280         BHI     INCDX   IF GREATER
01290         BLO     DECDX   IF LESS
01300         CLRA            IF THE SAME
01310         BRA     NDX1
01320 INCDX   LDA     #1      DX=+1
01330         BRA     NDX1
01340 DECDX   LDA     #-1     DX=-1
01350 NDX1    STA     500,Y   SAVE NEW DX
01360         LEAY    1,Y
01370 NEWDY   LDB     $C0     CURRENT Y COORDINATE
01380         CMPB    ,Y      COMPARE TO THE ARRAY
01390         BHI     INCDY
01400         BLO     DECDY
01410         CLRB            DY=0
01420         BRA     NDY1
01430 INCDY   LDB     #1      DY=+1
01440         BRA     NDY1
01450 DECDY   LDB     #-1     DY=-1
01460 NDY1    STB     500,Y   SAVE NEW DY
01470         LEAY    1,Y
01480         DEC     COUNT2  CHECK AGAINST ENTIRE ARRAY
01490         BNE     NEWDX
01500         LBRA    LOOP1
01510 DONE    RTS
01520 LOC     RMB     250*2
01530 DEL     RMB     250*2
01540 FINIS   NOP
01550         END     START
```

Product Review

CCTools: A Disk Manager in Need of Management

I'm always a little wary of disk-file managers because often they are loaded with bugs, and I've had several cause damage to the data on disks to the extent that I've had to rebuild the disks. With this in mind, I approached reviewing *CCTools* very carefully. My caution was reasonable, as I'll show below.

CCTools is a disk-management program that runs under OS-9 Level II. The system requires you have a minimum of 512K installed in your CoCo 3. The authors recommend a large disk (i.e., hard disk) to make all of *CCTools* features useful.

Even though *CCTools* is written in BASIC09, the program performed quickly. It does not require *Multi-Vue* and is considered a faster replacement for Tandy's GUI, although *CCTools* itself does not provide such a graphics environment.

Included with the package is a program designed to install *CCTools* on your system. However, the copy I received for review did not work properly. It appears that the files on the distribution disk are organized differently than the installation program expects. The *CCTools* documentation describes an alternative method for installing the system, but that approach also did not work. Since I am fairly familiar with OS-9, I was able to figure out where all the files were supposed to go and, finally, was able to get *CCTools* running.

When active, *CCTools* presents you with an attractive screen on which there is a menu bar and two large windows, each containing a listing of the current directory. You can use various key combinations to move through the directory trees easily and quickly. Several functions are provided on the menu bar that allow you to modify the screen view or file structure display. Other key combinations allow you to start programs, view file contents, copy files, etc.

CCTools offers the OS-9 user a tremendous amount of versatility. Most features can be customized to your preferences. You can select various programs that can be started from within the menu structure of *CCTools*, and you can even modify the way programs are used by *CCTools* to carry out file manipulations such as text editing and directory sorting. On the negative side, support for many of *CCTools*' functions is not built-in — you must supply external programs and other utilities to take advantage of them.

Among other things, the Analyze function provided with *CCTools* is supposed to rename directories so they follow the OS-9 standard and appear in all uppercase characters. However, it also renames all other files with lowercase characters and exhibits problems in distinguishing directories from other file types. This function indiscriminately renamed 90 percent of the files on my hard disk. Worse yet, once I started the function, I discovered there is no way to stop it.

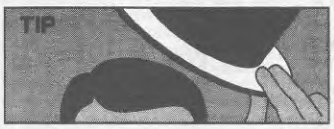
The *CCTools* documentation is included in the form of a file on one of the two distribution disks. Since this file is a standard text file, you can print it using just about any printer. On the other hand, as with other packages using this distribution approach, the documentation file is hard coded for 66-line pages. This presented some problem when I printed the file on a standard HP LaserJet, which supports only 60 lines per page.

The documentation is disorganized and does not clearly describe how to use the program. Several functions were described in different places in the manual, and these different references contradicted each other. Some functions, especially those requiring external programs, were given a brief mention only. For example, the only mention of the Clean function is that it "cleans directory entries." After my experience with Analyze, I didn't even try it.

It would have been nice were I able to contact someone from the Micro-80 Users Group for support in a timely fashion. Unfortunately, support for *CCTools* is available only through the mail. And it's a long way to Canada.

CCTools could, with some careful reworking, become a very useful tool for users of OS-9 Level II. However, without better documentation and provision for a safe way to back out of functions that have a global influence on disk structure, I cannot recommend the system. *(Micro-80 Users Group, 598 Riverton Avenue, Winnipeg, MB R2L 0P1, Canada; $15, $18 Cdn., money orders only.)*

— Bill Bodenholzer



Most CoCo users know that magnetism is what allows us to store large amounts of data on tape and disk. Many users also know, having found out the hard way, that the same phenomenon can also destroy that data in the wink of an eye. Magnetism giveth, and magnetism taketh away. Following are a few *dos* (and implied *don'ts)* for protecting data when you are storing tapes and disks:

— *do* keep tapes and disks at least six inches away from video monitors and televisions. Other devices to watch out for are tape recorders and audio speakers; just about any electromechanical device is capable of causing problems.

— *do* store tapes and disks well above the floor. One fellow kept his disk box on the floor and lost most of his software when his spouse came through with a vacuum cleaner — the motor in the powered head erased his disks.

— *do* keep tapes and disks *far* from "refrigerator" magnets. We know of one user who stored his daily backup by sticking it to the side of a filing cabinet with such a device from a pizza-delivery service. He couldn't understand why his data disk never worked. He has since outlawed refrigerator magnets in his home.

Feature Program

The CoCo Phone Book

by Trevor Boehm

CoCo Fone is a short program that keeps track of phone numbers, names and addresses. It requires a CoCo with at least 32K and Extended BASIC, and a disk drive.

When you first start *CoCo Fone*, my name, address and phone number are displayed. Also displayed is a menu of options. To select an option, just press its starting letter. For example, to remove my personal information, press D to select the Delete option. At this point, use the Add function to enter the appropriate information for your relatives, friends or business acquaintances.

When you finish entering the names and numbers, put a freshly formatted disk in Drive 0 and select Save. The file is written to disk under the name `PHONE#S.PHN`. Once the file is saved, you can load it for later use with the Load option.

As I mentioned earlier, pressing D to delete a record causes the entry currently being displayed to be blanked, effectively removing it from the list. Looking up a number is easy as well, requiring only that you enter a portion (or all) of the name of the person you want to find. To get a printed copy of the record currently displayed, press H to select the Hardcopy option. Finally, use the P (Previous) and N (Next) keys to step backward and forward through the entries.

I originally wrote *CoCo Fone* to keep track of fellow members of CoCo users groups. No doubt you will find many other uses for this program.

Trevor Boehm is a tenth-grade student whose greatest passion is challenging computers with new programs. He has participated in several science fairs and has received numerous awards for his work. He can be contacted at 77 Inwood Cres., Winnipeg, MB R2Y 1A2, Canada. Please include an SASE when requesting a reply.

32K Disk

The Listing: COCOFONE

```
1 'COCOFONE
2 'BY TREVOR BOEHM
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
9 'PALETTE 13,0:PALETTE 12,63
10 CLEAR 5000:DIM N$(100),A$(100
),B$(100),H$(100):HN=1:N$(1)="TR
EVOR BOEHM":A$(1)="77 INWOOD CRE
SC.":H$(1)="(204) 832-4495":B$(1
)="N/A":C=1:CC=1
20 CLS:PRINT:PRINTTAB(8)"COCO PH
ONE BOOK":PRINTTAB(8)"═══════
═══"
30 PRINT:PRINT"  NAME: ";N$(C)
40 PRINT"  ADDRESS: ";A$(C)
50 PRINT"  BUSINESS NUMBER: ";B$
(C)
60 PRINT"  HOME NUMBER: ";H$(C)
70 PRINT:PRINT:PRINT"MENU:":PRIN
T:PRINT"<A>DD <D>ELETE <C>HANGE
<S>AVE":PRINT"<L>OAD <F>IND <N>E
XT <P>REV":PRINT"<H>ARDCOPY <O>U
IT"
80 A$=INKEY$:IF A$="" THEN 80
90 IF A$="P" THEN C=C-1:IF C=0 T
HEN C=1:GOTO 20 ELSE GOTO 20
100 IF A$="N" THEN C=C+1:IF C>CC
 THEN C=CC:GOTO 20 ELSE GOTO 20
110 IF A$="O" THEN CLS:PRINT"TYP
E GOTO 20 TO RESUME":END
120 IF A$="A" THEN CC=CC+1:C=C+1
:CLS:LINEINPUT"NAME: ";N$(C):LIN
EINPUT"ADDRESS: ";A$(C):LINEINPU
T"BUSINESS NUMBER: ";B$(C):LINEI
NPUT"HOME NUMBER: ";H$(C):GOTO 2
0
130 IF A$="C" THEN CLS:LINEINPUT
"NAME: ";N$(C):LINEINPUT"ADDRESS
: ";A$(C):LINEINPUT"BUSINESS NUM
BER: ";B$(C):LINEINPUT"HOME NUMB
ER: ";H$(C):GOTO 20
140 IF A$="S" THEN CLS:PRINT"REA
DY DISK...":EXEC44539:OPEN"O",1,
"PHONE#S/PHN":FOR X=1 TOCC:PRINT
#1,N$(X):PRINT#1,A$(X):PRINT#1,B
$(X):PRINT#1,H$(X):NEXT:CLOSE:PR
INT"SAVED...":EXEC44539:GOTO20
150 IF A$="L" THEN CLS:PRINT"REA
DY DISK...":EXEC44539:OPEN"I",1,
"PHONE#S/PHN":FOR X=1 TO CC:LINE
INPUT#1,N$(X):LINEINPUT#1,A$(X):
LINEINPUT#1,B$(X):LINEINPUT#1,H$
(X):NEXT:CLOSE:PRINT"LOADED...":
EXEC44539:GOTO20
160 IF A$="F" THEN CLS:LINEINPUT
"NAME: ";F$:FOR X=1 TO CC:IF INS
TR(N$(X),F$)<>0 THEN C=X:GOTO 20
 ELSE NEXT:PRINT"NOT FOUND...":E
XEC 44539:GOTO20
170 IF A$="D" THEN N$(C)="":A$(C
)="":B$(C)="":H$(C)="":GOTO20
180 IF A$="H" THEN CLS:PRINT"REA
DY PRINTER...":EXEC44539:PRINT#-
2,"NAME: ";N$(C):PRINT#-2,"ADDRE
SS: ";A$(C):PRINT#-2,"BUSINESS N
UMBER: ";B$(C):PRINT#-2,"HOME NU
MBER: ";H$(C):GOTO20
190 GOTO 80
```

OSK FROM COVER

I ordered a complete system consisting of the circuit boards, 3 megabytes of memory, an XT-compatible keyboard, a 3½-inch high-density floppy drive, and a case and power supply. I received everything but the I/O board. This board is on back order, and I hope to receive it soon; the SCSI hard-drive host adapter is on the I/O board, and I cannot yet use a hard drive with the system. To offset this, I added a second 3½-inch high-density drive. [*Editor's Note: Mr. Donaldson wrote this article prior to receiving the I/O board for his MM/1. While the article was in production at THE RAINBOW, however, he received the board and sent us an update. We decided to publish the update separately. It will appear with Part 2 of this article in the December 1992 issue.*]

So far I have been very pleased with the MM/1, and the software that comes with it is impressive. Included are Microware's C compiler and *Microware Basic*. The latter is the same as BASIC09 for the CoCo except there is no graphics library (`gfx` or `gfx2`) at the present time. A `cfgx.1` graphics library is included for the C compiler. This graphics library is nearly identical to the version of `cgfx.1` that comes with the OS-9 Level II *Development System*. The main differences are features that allow you to use the more powerful features of the 68070 CPU and the VSC graphics coprocessor.

Using the MM/1 as a floppy-based system reminded me of the days on my CoCo 3 before I got a hard drive. The first thing I did was create three system disks: one for C programming, one for BASIC programming, and one for telecommunications.

When I tried to run a program on my MM/1 for the first time, I received an error telling me that OS-9 could not find `/TERM`. I soon discovered that the OS-9/68000 uses a file called `TERMCAP` that tells it what kind of terminal you are using. The terminal identifier for the MM/1 is called `VSC`. So I used OSK's environment variables to set the terminal type as follows:

```
setenv TERM vsc
```

Once OSK knew my terminal parameters, programs like *Sterm* and *MicroEmacs* ran fine.

The purpose of the `TERMCAP` approach is to allow software to be used with many different terminals. For those who use OS-9 on the CoCo, *Dynacalc*, *Scred* and *DynaStar* use a similar file. Since OS-9 is a multiuser operating system, `TERMCAP` makes it easy to have different types of terminals connected to different serial ports. Different terminals have different parameters for handling functions such as the number of display lines, function-key definitions, cursor controls, etc. `TERMCAP` redefines these functions so that different terminals look the same to the operating system.

I soon got tired of entering my terminal setting when I booted the system, so I looked for a better way to handle that task. From the OS-9 manuals, I learned that OS-9/68000 supports two unique files named `.LOGIN` and `.LOGOUT`. (Notice that there is a period in front of these filenames; this tells OSK that the file is invisible — the names won't show up in a standard directory listing. Appending a period to the beginning of any file or directory name hides it from view.) The entries in these two files affect the user's environment. I added the following environment settings to my `.LOGIN` file:

```
setenv TERM vsc
setenv MODEM /t0
setenv PROMPT MM/1:
xmode /t0 baud=1200 type=0
```

Now OS-9 knows what my terminal type is, which serial port I want to use for my modem, what I want the prompt to look like (instead of OSK's default dollar sign), and the serial-port settings I want.

The `.LOGOUT` file works the same way except that its contents are executed when you enter `logout`. This is useful in multiuser setups and for simply returning the environment to its default status. `logout` is also used to terminate a window or shell, much like the `ex` command in OS-9 Level II. If you try to enter `ex` in an OS-9/68000 shell, you'll receive an error message telling you that you can't log out that way.

Let's go back to the hidden files for a minute. If a file is invisible, how can you tell if it's stored in a directory? OSK's `dir` command provides many more features than Level II `dir` command. To see hidden files, you enter the `-a` option on the command line. Another option is `-u`, which tells `dir` not to format the listing. OSK also supports wildcards.

These features allow you to enter such fancy command lines as

```
dir -u *.c ! attr -z
```

The `-u` tells `dir` to display each directory entry on a separate line, and `*.c` tells it to list only those entries that end with a `.c`. Then the output of the `dir` command is piped through `attr` so the attributes for the specified files are also listed.

With OS-9 on the CoCo 3, I used `cobbler` instead of `OS9Gen` to create new boot disks because it was much easier. `cobbler` is not included with OS-9/68000. Instead, Microware made the `OS9Gen` utility very easy to use. Interactive Media Systems includes several script files for creating boot disks. Since I haven't received the I/O board, all I had to do was alter a script file so that it didn't add the sound and mouse drivers to the new boot file. After this, making a boot disk was easy — I simply entered `boot.1mfd`, and the script file assembled the modules and built a boot file on the disk in Drive `/d1`.

Another easy way to create boot disks is to copy the boot file from `/d0` to `/d1`. Then just enter

```
os9gen /d1 -eq=/d1/os9boot
```

The options `-eq` are used when linking to LSN0. The `-e` option tells the system to use an extended boot size (larger than 64K), and `-q` stands for "quick boot." It links LSN0 to point to the starting address of the boot file. The boot ROMs will then load the system from the `OS9Boot` file. Since `OS9Boot` is *not* limited to 64K, I like to make a master disk and just use this method to make boot disks. Another handy thing to know is that, unlike `OS9Boot` with Level II, the boot file doesn't have to fit into a continuous space.

Updated modules and drivers for the MM/1 and OSK have been posted on Delphi and CompuServe in the OS-9 SIGs. I recommend you check these updates and, if they are newer than what you received with your system, download the archive. After I stored the unarchived modules on my Bootmods disk, I created a new boot file. The first thing I noticed when booting after that is the boot file loads and executes much quicker. Disk access is also somewhat faster. Another feature is that the console bell now beeps (it seems there was a bug in the early `windio` driver).

When setting up a startup file for OSK, make sure the first line is something like

```
iniz /d0 /d1 /r0 /t0 /t1 /p
```

When booting a floppy-based system, it isn't necessary that you initalize the drivers using `iniz`. However, it is required for hard-drive systems, and it makes good sense to go ahead and do it on floppy systems. This way they are at the top of memory and their buffers have already been initialized. It also keeps the system from fragmenting memory. I always did the same thing in my startup file on the CoCo 3. There, if you didn't initialize the floppy-disk drivers, the drive would sometimes stay on after the system switched to the hard drive.

Next month I'll discuss some other features OSK offers and share my initial experiences with the OS-9/68000 C compiler.

John Donaldson is a software engineer in the gas-turbine division of Stewart & Stevenson Services. In addition to working with computers, his hobbies include amateur radio, model railroading and sailing. His Delphi username is VAXELF.

DELPHI BUREAU

EDDIE KUNS

Using the Forums

Last month I introduced basic information about Delphi Forums. This month, let's build on that foundation.

So what is a thread? Basically, all replies to a given message, and replies to these replies, constitnte a thread. A thread is a continuing discussion. You can tell a message is part of a thread if, after reading the message, you see the message `Enter FOLLOW for related Message(s)`. If you type `FOLLOW` at the `FORUM>` prompt, you are shown the next message in the thread. You need only press ENTER after each message to see the following message in the thread; between messages you will see `Press RETURN for related Message(s)`. When you have read all of the following messages in the thread, you see the message `End of Thread. Press RETURN to resume`. At this point, pressing ENTER brings you to the first available message following the message where you typed `FOLLOW`.

The reverse of the `FOLLOW` command is `BACK`. If you enter `BACK`, you see the thread in reverse. First you see the message the current message is in reply to. The next time you press ENTER you see the message which that message was in reply to, etc. You won't necessarily see every message in a thread when you use this command.

The Forum software keeps track of all messages you have read, but only until you leave Forum. Thus, if you follow a thread and then exit Forum, you may set your high-message counter higher than you want. The high-message counter is set to the highest message number yon read, even thongh you have not read all intervening Forum articles. If you don't want to skip messages like this, you can use the `HIGH` command to set yonr high-message connter to make sure that you don't lose any messages. Here is an example:

```
FORUM> high

Last Message in Forum: 63287
High Message on Entry: 62806
Current High Message : 63215
New Value (or RETURN):
```

I can type the number I want my high-message counter set to, or I can just press ENTER to keep my current setting.

If you enter `READ NEW` right before exiting Forum, you are presented with the lowest numbered new message you haven't read in this Forum session. A new message, of course, is a message higher numbered than your high-message counter when you entered Forum. Thus, if you `READ NEW` and then set your high-message counter to the message number of that message, you won't miss any Forum messages.

Now that you know what a thread is, how do you add your own thoughts to a thread? If you have just read the message to which you want to reply, simply enter `REPLY`. You can also type `REPLY` *article-number*, where *article-number* is the number of the Forum message you want to reply to. Either of these commands drop you into Delphi's simple line editor. Enter your reply and press CTRL-Z or `/EXIT` when you are finished. You can also enter `/EDIT` to enter your chosen online editor, `/LIST` to list the lines you have entered so far, and `/DEL` to delete the last line of your article.

If you have uploaded a response into yonr workspace — thus doing the editing offline and saving yourself some money — enter `REPLY` *article-number filename* to enter yonr reply. Don't forget to delete the file from yonr workspace when you are finished! You can also enter `REPLY MAIL` to send a private reply via Mail.

If you want to start a thread — say yon have a question you want to ask — use the `ADD` command. Simply enter `ADD` and answer all the prompts. You need to decide who the message is to (the default is `ALL`), the message topic, and the message's subject. Be careful to make the subject meaningful. After entering `ADD`, use Delphi's simple line editor described above.

What do you do when you see a lengthy thread in which you are completely uninterested? You certainly don't want to spend the time online reading through it. If you enter `IGNORE THREAD`, Forum remembers to not show you any articles which belong to the same thread as the current article.

The `IGNORE` command is much more powerful than this, however. You can also enter `IGNORE SUBJECT AUTO` to ignore all messages with `AUTO` in the subject. If there are certain people you don't want to hear from, you can enter `IGNORE FROM` *name* where *name* is the Delphi username of the person you want to ignore. `IGNORE` settings remain in effect only until you leave Fomm or change them. `IGNORE NONE` turns off all `IGNORE` settings.

Two more very useful commands are + and -. If you just finisbed reading Forum message 63035, entering +5 shows you message 63040. Of course if message 63040 has been deleted, it shows a following message. You could also have typed - 2 to see Forum Message 63033. Entering only - is equivalent to entering `CURRENT` — the current message is redisplayed. One nice thing about the - command is that it does not interrupt the flow of `READ NEW`; when you press ENTER by itself, you see the next *new* message. In contrast, if instead of - 2 you enter 63033, pressing ENTER shows you message 63034 next.

If someone has sent you a "See you next Tuesday" message in Forum, and you decide that it doesn't need to stay around any longer, you can use the `DELETE` command to delete the Forum message. You can delete any message you have posted to Forum and any message posted to you. Be careful: once a message is deleted, no one can get it back.

The appropriate final command to describe this month is `LAST`. Just as in mail, `LAST` displays the highest-numbered message. Remember, this sets your high-message counter to the highest message in Forum! If you want to read some of the prior messages, you may want to use the `HIGH` command before leaving Forum (or the next time you enter Forum if you wrote down your desired high-message number).

Next month I'll finish describing Forum, touching on some more advanced commands.

June Uploads

In the OS-9 SIG General Information database, **Tim Kientzle** (TIMKIENTZLE) released a file containing MM/1 expansion information, including information about monitors, keyboards, SCSI hard disks and mice that are known to work with the MM/1. He has also included useful hardware information. **Cray Augsburg** (CRAY) uploaded THE RAINBOW's submission guidelines. **Jim Vestal** (JIMVESTAL) submitted Allen Huffman's report on the First Annual Last Cocofest. **Chris Burke** (COCOXT) explains how to install a new 6309 CPU in your CoCo without physically removing your 6809 — something many may want to do when installing their PowerBoost, which **Marty Goodman** (MARTYGOODMAN) reviews. Chris Burke also released a patch to `booster` and information to help those who get Error 221 when trying to run `booster` — both in the

System Modules (6809) database. Finally, **Alfredo T. Santos** (ALFRADIO) released a history of the CoCo.

In the Telecom (6809) database, **Jim McDowell** (JMLSOFT) released *AutoQuest* — a program useful when playing Delphi's *Quest for the Holy Grail* found in the Entertainment section. The program uses a split screen and is a front-end to Delphi's game. **Hugo Bueno** (MRGOOD) uploaded Virginia's UUCP map for someone who wanted to find a UUCP connection. Files like this appear from time to time in the databases but don't necessarily remain for years, as the information in them changes rapidly.

Joel Hegberg (JOELHEGBERG) released a couple of useful utilities into the OSK Applications database — both run under K-Windows on the MM/1. *WSee* allows you to look at the contents of another window while *SnapShot* allows you to save the contents of another window to disk and view it later. **Leonard Cassady** (MAUDIB) uploaded a patch that customizes *Uqvt70* for *DynaStar* on the TC-70.

Mark W. Farrell (XLIONX) wrote a quite extensive review of methods of applying patches. This is a companion to his prior upload `NewLev2.ar`, listing a great many known patches and enhancements to OS-9 Level 2 and applications. If you have been afraid to apply patches, fearing it would be too difficult or too dangerous, read this!

In the CoCo SIG, **Allen Huffman** (SUBETHA) released a report of the First Annual Last CoCoFest at Chicago. Cray Augsburg (CRAY) released THE RAINBOW's submission guidelines. These two groups are the same as those in the OS-9 SIG. Also duplicated is Alfredo Santos' (ALFRADIO) history of the Color Computer.

In Utilities & Applications, **Joe Sannucci** (SANNUCCI) uploaded the final version of *Japan123* by Larry Greenfield. This program allows you to edit Japanese text, and print it to a dot-matrix printer. **Erich Schulman** (ESCHULMAN) released a program which allows viewing `.MAX` files on an IBM PC-type computer. **John Saya**'s (SAYA) *Switch* program converts an ASCII data file into a BASIC file you can load and run. When you run the switched file, you see the original text file!

Marty Goodman (MARTYGOODMAN) reviews the Burke & Burke PowerBoost in the Product Reviews & Announcements database. **Alvin Cotton** (ACOTTON) released the StarLink BBS in the Telecommunications database.

Eddie Kuns is pursuing a doctorate in physics at Rutgers University. He lives in Aurora, Illinois, and works as a programmer and researcher at Fermilab. Eddie is the database manager of the OS-9 SIG and can be reached online as EDDIEKUNS.

DATA BASE REPORT

OS-9 SIG

General Information
- `MM1 EXPANSION INFORMATION` — TIMKIENTZLE — Tim Kientzle
- `CALL FOR PROGRAMMERS` — NIMITZ — David Graham
- `RAINBOW SUBMISSION GUIDELINES` — CRAY — Cray Augsburg
- `COCOFEST REPORT BY ALLEN HUFFMAN` — JIMVESTAL — Jim Vestal
- `6309 INSTALLATION INSTRUCTIONS` — COCOXT — Christopher Burke
- `COCO CHRONICLES` — ALFRADIO — Alfredo T. Santos
- `INFO ON POWER BOOST` — MARTYGOODMAN — Marty Goodman

Applications (6809)
- `TRIM BYTES FROM FILES` — ESCHULMAN — Erich Schulman
- `GOLFHCP: GOLF HANDICAP UTL.` — RAYMAYEUX — Raymond Mayeux
- `MDIR 4.00: MODULE DIR. UTILITY` — MITHELEN — Paul Jerkatis

Telecom (6809)
- `AQTERM MINI TERMINAL PROGRAM` — JMLSOFT — Jim McDowell
- `AUTOQUEST BUFFER INSTRUCTIONS` — JMLSOFT — Jim McDowell
- `AUTOQUEST VERSION 1.0` — JMLSOFT — Jim McDowell
- `VIRGINIA UUCP MAP` — MRGOOD — Hugo Bueno

System Modules (6809)
- `BOOSTER ERROR 221 INFO` — COCOXT — Christopher Burke
- `FIXBOOSTER: PATCH FOR POWERBOOST` — COCOXT — Christopher Burke

Games & Graphics
- `KSCOPE RAYTRACE (GIF)` — JOHNBAER — John Baer

Programmers Den
- `GIFTOPPM CONVERTER` — THEFERRET — Philip Brown

OSK Applications
- `WSEE FOR THE MM/1` — JOELHEGBERG — Joel Hegberg
- `SNAPSHOT FOR THE MM/1` — JOELHEGBERG — Joel Hegberg
- `TEXPRF: TEXINFO TO PROFF` — JOHNREED — John Wainwright
- `GRAPHICAL CLOCK FOR MM/1` — JOELHEGBERG — Joel Hegberg
- `SH DOCUMENTATION` — THEFERRET — Philip Brown
- `UQVT70 MODIFIED FOR DYNASTAR` — MAUDIB — Leonard Cassady

OSK Telecom
- `PHILMAIL: UUCP OFFLINE MAIL READ` — THEFERRET — Philip Brown

Tutorials & Education
- `HOW TO SAFELY DO PATCHES` — XLIONX — Mark W. Farrell

COCO SIG

General Information
- `COCONUTS OF AMERICA CLUB` — BFLAHIVE — Brian Flahive
- `LAST FEST '92 REPORT` — SUBETHA — Allen Huffman
- `RAINBOW SUBMISSION GUIDELINES` — CRAY — Cray Augsburg
- `ARCHIVED COCO CHRONICLES` — MARTYGOODMAN — Marty Goodman
- `THE "COCO" CHRONICLES` — ALFRADIO — Alfredo T. Santos

CoCo 3 Graphics
- `COLOR DEMO` — BFLAHIVE — Brian Flahive
- `BEATLES IN CM3` — DEANHOLDER — Dean Holder
- `STAR TREK:TNG RASCAN IMAGES` — DEANHOLDER — Dean Holder
- `BOUNCIN' BALL` — SANNUCCI — Joe Sannucci

Utilities & Applications
- `JAPANESE-WRITER PROGRAM` — SANNUCCI — Joe Sannucci
- `VIEW .MAX IMAGES ON A PC` — ESCHULMAN — Erich Schulman
- `SWITCH - UPDATED` — SAYA — John Saya
- `SWITCH` — SAYA — John Saya

Games
- `COOKIE MONSTER TO DISK` — MARTYGOODMAN — Marty Goodman

Product Reviews & Announcement
- `INFO ON POWER BOOST` — MARTYGOODMAN — Marty Goodman

Telecommunications
- `STARLINK BBS` — ACOTTON — Alvin Cotton

Product Review

GrafExpress 2.0: On the Road to Multimedia

Multimedia on a CoCo? That's a pretty tall order for an 8-bit computer — even a machine as powerful as my beloved CoCo 3. So it was with some skepticism that I read Sundog Systems' advertisement for *GrafExpress 2.0*. When I received the product for review, I got a chance to see if that skepticism was warranted.

GrafExpress is a library of software routines that allows you to create your own high-speed, high-resolution graphics programs. It also includes support for four-voice sound. This library gives you the tools you need to write, say, the next great arcade game or perhaps a user-friendly application complete with a fantastic GUI. *GrafExpress* doesn't add commands *to* BASIC but works *with* it. You can write programs in BASIC or assembly language (an assembler is not included), or a combination of the two.

GrafExpress requires a CoCo 3 and a disk drive. Although the package works fine with 128K, you'll find 512K adds to it a great deal of flexibility.

Graphic images can be generated through typical drawing commands such as box, line, rectangle, circle, color, set and fill. These commands can address locations that extend beyond the viewing area, and that portion of the drawing that is visible will be displayed correctly through the window. The graphics area may be scrolled to reveal different parts of the drawing. Graphics can also be loaded from disk and displayed.

GrafExpress gives the CoCo 3 programmer the ability to use sprites — "magical" little graphics elements that can be moved around the screen, passing under or over other images. Sprites can be generated using drawing commands or loaded from disk, and two types are supported: "sticky" sprites leave a copy of themselves when they are moved, while "nonsticky" sprites move around the screen without changing the background. Sprites are assigned a numerical value for identification, and functions are included that tell the system if a sprite comes in contact with another sprite or range of sprites. Stationary sprites can be used to define boundaries for a moving object — by checking for a collision, the program knows if the sprite has moved into the boundary. Collisions between moving sprites can also be tested.

GrafExpress allows you to create and manage windows. These windows can cover the entire screen or occupy only a portion of it. They may overlap, but *GrafExpress* does not save the image under a window. You must preserve the image yourself if you'll need to restore it later. Alternatively you can rewrite the entire window after the covering window is closed. Each window can function independently of other windows and may contain completely different images or information. Windows can also work as a virtual viewport into a complete image, allowing various parts of the image to be seen.

Text can be displayed onscreen in a variety of ways. Font width and height are easily altered, and characters can be displayed with a solid background color or may be transparent, allowing the background to show through. Scrolling text on a graphics screen is also easy to accomplish.

GrafExpress' music commands provide programmers with the ability to work with up to four voices at one time. A sound's envelope (the way the sound starts, sustains, and dies away) and timbre (e.g., brass, flute or any imaginable quality) can be controlled. Although pitch is specified through a non-musical system of numbers, the system provides the ability to generate a greater variety of sounds; and the documentation includes suggestions for creating special sound effects. A table for converting the numerical values to musical notes is also included. Other sound commands allow you to change the tempo and duration of a sound.

Finally, *GrafExpress* has commands that provide joystick support for systems with and without the Tandy Hi-Res adapter. The clock speed of the CoCo 3 can also be controlled using a simple command.

Programs written in BASIC call *GrafExpress* by first setting up a string variable that includes the commands to be passed then executing *GrafExpress*. The program works with integer variables so that numerical values can be updated as it runs. This may cause some difficulty since BASIC on the CoCo is always in the floating-point mode. But if you remember that you must remove any part of a number to the right of the decimal point (using FIX or INT), you shouldn't have any trouble. Several examples of how to write programs using *GrafExpress* are included on the distribution disk, and the documentation clearly explains each function of *GrafExpress*. When writing BASIC programs for use with the system, you will find that the command syntax for *GrafExpress* is very similar to that supported by Disk BASIC.

Machine-language programs call the *GrafExpress* routines directly by setting up the necessary registers and executing a JSR to *GrafExpress*. Sundog Systems has included files that make getting started easy. By including the assembly-language files in your own programs and initializing *GrafExpress*, your machine-language creations can benefit from the graphics and sound routines available with *GrafExpress*. The examples included on the *GrafExpress* disk are well-commented and make it easy to write your own programs.

The documentation for *GrafExpress* is in the form of a typeset 48-page booklet. Its four sections cover various aspects of the system and include detailed explanations of all of the instructions. The first section explains in general terms the function of the system and how the remainder of the booklet is organized. This section also explains some of the philosophy behind the development of *GrafExpress*. This information is helpful when writing programs since it helps you understand how the instructions interact with your computer.

The second (and largest) section gives detailed explanations of each *GrafExpress* function. The information is organized logically according to the type of function. Specific examples regarding the general use of each instruction are included, as are tips for more efficient use of the instruction.

Section 3 is intended for those who want to use *GrafExpress* with assembly language. It explains how to include the source-code files on the disk, making the use of *GrafExpress* functions in your programs much easier.

The fourth section covers the use of the system's auxiliary programs. The *GrafExpress* distribution disk includes several auxiliary programs that help you create your own programs. *Pic-Maker* provides an easy way to build sprites in any of the supported resolutions. *Wave-Maker* is a neat application you can use to generate sound waves for use with the sound functions. I was able to create sound waves and envelopes with *Wave-Maker* and was impressed with its appearance and function. Onscreen buttons make it easy to test sounds under different conditions. *ArtExpress* is a simple graphics editor that allows you to create pictures in the 256-color mode (256 colors are available only with a color composite video monitor or a color television). All of these applications were written using *GrafExpress* functions. Probably the best part about these applications is the fact that the source code is included on the disk. This allows you to take a look at how programs are written for *GrafExpress*. Other utilities allow you to import portions of any HSCREEN2 graphics screen and CoCo 3 fonts for use with *GrafExpress*.

Although there is no index in the manual, the table of contents includes enough entries that readers are quickly guided to the desired information. A single errata sheet explained the use of some of the newer utilities and also gave information on how to patch *GrafExpress* for use with ADOS.

If you write an application that relies on *GrafExpress* routines, you can legally distribute only those portions of the program that you have written. If another person wants to run your application, he must have his own copy of *GrafExpress* for the program to work. Alternatively, Sundog Systems offers licensing agreements under which you may distribute your application along with those parts of *GrafExpress* needed to make your program functional.

In my time with the product, everything appeared to function as it should. As expected, the assembly-language programs performed faster than the BASIC programs, but even the BASIC programs ran with incredible speed. There is certainly nothing wrong with that. The only "fault" I noticed was that the cursor kept jumping to the right side of the screen when I used the Hi-Res joystick adapter. This may need a little tweaking on the part of Sundog Systems. Although the system works with 128K, you'll be able to use more sprites, windows and screens — as well as longer music — if you have 512K in your CoCo 3. In other words, at least in this case, more *is* better.

The documentation was far better than much of the documentation I've seen, and the supporting programs and programming examples make it easy for the average programmer to write his own applications. And more-experienced programmers should be able to write some really fantastic games and business and graphics applications using *GrafExpress*. *(Sundog Systems, 21 Edinburg Drive, Pittsburgh, PA 15235, 412-372-5674; $34.95 plus $2.50 S/H).*

— Bill Budenholzer

CELSIUS FROM COVER

other. As written, the program works on the CoCo 3, though it should be fairly easy to modify for nse with a CoCo 1 or 2.

The relationships between the Fahrenheit and Celsius temperature scales are given by two simple equations (actually one equation that can be inverted depending on the conversion desired). As shown in Line 210 of CELSIUS, the equation for converting from Fahrenheit to Celsius is

$$C = \frac{(F-32)*5}{9}$$

where C and F are the Celsius and Fahrenheit values, respectively. Rearranging terms, if we know the Celsius value, we can determine the corseponding Fahrenheit value using

$$F = \frac{(9*C)}{5} + 32$$

This formula appears in Line 290, though I use different names for the variables involved.

It is things like this that make using the CoCo so much fun. I hope your curiosity is similarly piqued by the little things in life.

Roger Carlson is a quality-assurance consultant in the Chicago area. In addition to using his CoCo, he enjoys fishing, camping and being outdoors. Roger's Delphi username is PERCH.

CoCo 3

The Listing: CELSIUS

```
1 'CELSIUS
2 'BY ROGER L. CARLSON (PERCH)
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
20 REM FORMULA FOR FAHRENHEIT TO
 CELSIUS IS C=(F-32) * 5/9
30 REM FORMULA FOR CELSIUS TO FA
HRENHEIT IS F=C * 9/5 + 32
40 REM IF YOU WANT TO GO HIGHER
THAN 99,999.9 DEGREES, THEN ADD
[#] SIGNS TO THE PRINT USING COM
MANDS.
50 POKE&HE03E,36
60 WIDTH40:CLS
70 PALETTE8,255:PALETTE0,0
80 LOCATE12,10:PRINT"TEMPERATURE
"
90 LOCATE4,12:PRINT"CONVERT FAHR
ENHEIT TO CELSIUS"
100 LOCATE6,14:PRINT"AND CELSIUS
 TO FAHRENHEIT"
110 LOCATE0,22:PRINT"PRESS ANY K
EY"
120 EXEC34442:GOTO 130
130 CLS:LOCATE18,8:PRINT"MENU"
140 LOCATE13,10:PRINT"1) FAHRENH
EIT"
150 LOCATE13,12:PRINT"2) CELSIUS
"
160 LOCATE13,14:PRINT"3) QUIT"
170 LOCATE0,22:PRINT"SELECT 1-3"
180 AN$=INKEY$:IF AN$=""THEN 180
190 IF AN$="1"THEN200 ELSE IF AN
$="2"THEN 270 ELSE IF AN$="3"THE
N 350
200 CLS:LOCATE12,11:INPUT"FAHREN
HEIT";F
210 C=(F-32)*5/9
220 CLS
230 LOCATE8,10:PRINTF"DEGREES FA
HRENHEIT ="
240 LOCATE7,12:PRINTUSING"#####.
#";C;:PRINT" DEGREES CELSIUS"
250 LOCATE0,22:PRINT"PRESS ANY K
EY"
260 EXEC34442:GOTO 130
270 CLS
280 LOCATE12,10:INPUT"CELSIUS";C
E
290 CL=CE*9/5+32
300 CLS
310 LOCATE8,10:PRINTCE"DEGREES C
ELSIUS ="
320 LOCATE7,12:PRINTUSING"#####.
#";CL;:PRINT" DEGREES FAHRENHEIT
"
330 LOCATE0,22:PRINT"PRESS ANY K
EY"
340 EXEC34442:GOTO 130
350 POKE&HE03E,18
360 PALETTE RGB
370 WIDTH32
380 END
```

Feature Program

Echo ((revisited))

by John Collyer

While going through my back issues of THE RAINBOW, I came across a modified version of the OS-9 `echo` command, written by Stephen Goldberg ("KISSable OS-9," March 1989, Page 153).

This version gives you some Unix-style enhancements that allow certain types of screen control from within `echo`, making it far more useful. Since I was lucky enough to get a copy of OS-9 Level I, which includes and assembler, I decided to try assembling this handy utility.

As I entered the source code to Mr. Goldberg's utility, I thought it might be nice to have a way to repeat characters with the `echo` command. This would make creating menus and the like much easier. I added the code to do this, and the version of `echo` printed here includes this code as well as a little hexadecimal feature from Steve.

The new functions are documented in the source code (Listing 1), and the original functions are outlined in the March 1989 issue. However, I have provided a brief overview (see Figure 1). To generate a backslash in OS-9, hold CTRL and press the slash key. As you use the program, remember not to enter any numbers greater than 255, and make sure you put a comma between the parameters for the `\r` option. Also remember the echo output string holds only 254 characters, so keep it at or under that number. This shouldn't be too difficult since you can always enter another `echo` command line to fill in as needed.

Those users without an OS-9 assembler can enter the BASIC09 procedure shown in Listing 2. `MakeEcho.b09` generates the executable `echo` program. If you are using BASIC09 under Level I, however, make sure you change all references to `/dd` to another device (such as `/d0` or `/h0`) that is supported under Level I.

If you are using OS-9 Level II, the standard `echo` is loaded as a part of `shell`. For this reason, you may want to change the name echo to something else. This is easy to do with the assembly version, but the BASIC09 version requires a couple of changes (though still easy to make). To change the name to `echi` with `MakeEcho.b09`, change all occurrences of `echo` to `echi` in the listing. Then change the value 239 in the second `DATA` statement to 233.

John Collyer has been programming since he got his first Color Computer in 1987. He can be contacted at 323 W. Union St. #6, Medina, OH 44256. Please include an SASE when requesting a reply.

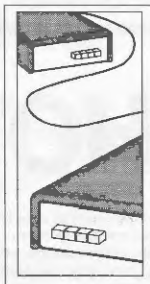
`\n`	go to a new line (linefeed)
`\c`	terminate display without a linefeed
`\f`	clear the screen (form feed)
`\\`	print a backslash
`\xxx`	print the character with an ASCII value of *###*
`\$xx`	print the character with a Hex ASCII value of *##*
`\rxxx,yyy`	print the character represented by *xxx* in ASCII *yyy* consecutive times
`\r$xx,$yy`	print the character represeuted by Hex *xx* in ASCII *yy* consecutive times

Figure 1: New Options for `echo`

OS-9

Listing 1: `Echo.asm`

```
*******************************************
* ECHO - COPYRIGHT (C) 1988 by S.B.GOLDBERG
*
* New Hex conversion function:
*     \$## - print hex ## ASCII character
*
* New Repetition function (C) 1991 by J.R.COLLYER
*     \r###,###  or  \r##,$## - print ASCII, number of times
*
          ifp1
          use /d0/defs/os9defs
          endc
*
          mod len,name,prgrm+objct,reent+1,entry,dsiz
*
hicount   rmb   1          msb character count
locount   rmb   1          lsb character count
dcount    rmb   1          digit count
rcount    rmb   1          repetition count
buffer    rmb   255        maximum length
          rmb   200        stack
          rmb   200        parameter
dsiz      equ   .
*
name      fcs   /Echo/
          fcb   4          edition number
          fcc   /(c)1988 S.B.Goldberg/
*****************************
* GENERATE DISPLAY
*****************************
entry     clr   hicount    zero character count
          clr   locount
          leay  buffer,u   start of text
          pshs  y          save parameter pointer
loop      ldb   ,x+        get text character
          cmpb  #'\        backslash?
          bne   save       no, save character
          ldb   ,x+        yes, get next character
          cmpb  #'\        backslash?
          beq   save       yes, save it
          cmpb  #'$        hex digits?
          bne   makelow    no, continue
          bsr   hextobin   hex number to binary
          bsr   checknum   check for hex digits
          bra   save       save it
makelow   orb   #32        make lower case
          cmpb  #'c        print without carriage return?
          beq   print      yes, print line
          cmpb  #'f        form feed (clear screen)?
          bne   newln      no, check for new line
          ldb   #$0c       yes, clear screen character
          bra   save       save it
newln     cmpb  #'n        new line?
          bne   repeat     no, check for repetition
          ldb   #$0a       yes, line feed
          bra   save       save it
repeat    cmpb  #'r        repetition needed?
```



INTERCOM

Pen pals

I have a 128K CoCo 3 with an FD-502 disk drive, a DMP-106 printer and a color TV. I would like to hear from pen pals between the ages of 8 and 12, but I will write to people of all ages.

Kevin Smith
1958 Washington Avenue
Portland, ME 04103

CoCo clubs

CALIFORNIA

☎ StG Net West, Alan Sheltra, P.O. Box 38713, Hollywood, 90038, (818) 761-4135, BBS (818) 761-4721

COLORADO

☎ Colorado Springs Color Computer Club, Bud Ward, 1118 Claiborne Road, Colorado Springs, 80906-5513, (719) 392-8268

CONNECTICUT

☎ Connecticut CoConut Connection, Charles Joseph

CONTINUED ON PAGE 22

```
        bne   decimal     no, must be decimal number
        clr   rcount      zero repeat count
        bsr   hexordec    get repeat character
        bsr   checknum    check for digit
        pshs  b           save it
        ldb   ,x+         next character
        cmpb  #',         seperator?
        bne   error       no, exit
        bsr   hexordec    yes, repeat count
        puls  a           output character
        bsr   checknum    digits entered?
reloop  sta   ,y+         save it
        inc   locount     count it
        decb              done yet?
        bne   reloop      no, go again
        bra   loop        continue looking
hexordec ldb  ,x+         next character
        cmpb  #'$         hex digits?
        bne   dec         no, dec digits
        bsr   hextobin    yes, convert hex to bin
        rts               return
dec     bsr   dectobin    decimal number to bin
        ldb   ,y          get binary value
        rts               yes, return
checknum tst  dcount      digits entered?
        beq   loop        no, continue looking
        rts               yes, return
decimal bsr   dectobin    decimal to binary
        ldb   ,y          get binary value
        bsr   checknum    digits entered?
save    stb   ,y+         save output character
        inc   locount     count it
        cmpa  #$0d        end of parameter?
        bne   loop        no, get next character
print   ldy   hicount     output length
        puls  x           output address
        lda   #1          standard output path
        os9   i$writln    print it
        bcs   out         exit with error
error   clrb              clear error flag
out     os9   f$exit      quit

**************************
* CONVERT HEX TO BINARY
**************************
hextobin clrb             zero value
        clr   dcount      zero digit count
        bsr   hexdigit    convert digit
hexdigit lda  ,x          get character
        suba  #'0         make binary
        cmpa  #9          more than 9?
        bls   digisave    no, continue
        lda   ,x          get character
        anda  #$df        convert to upper case
        suba  #'0+7       make alpha character binary
        cmpa  #15         valid digit?
        bhi   back2       no, return
        cmpa  #10         valid digit?
        blo   back2       no, return
digisave pshs a           yes, save it
        lda   #16         multiply old
        mul               total by 16
        addb  ,s+         add current digit
        leax  1,x         bump pointer
        inc   dcount      count digit
back2   rts               return
**************************
* CONVERT DECIMAL TO BINARY
**************************
dectobin leax -1,x        reset pointer
        clrb              zero value
        clr   dcount      zero digit count
        bsr   decdigit    first digit
        bsr   decdigit    next 2 digits
decdigit lda  ,x          get digit
        suba  #'0         make binary
        cmpa  #9          valid digit?
        bhi   back        no, return
        pshs  a           yes, save it
        lda   #10         multiply old
        mul               total by ten
        addb  ,s+         add current digit
        adca  #0          greater than 255?
```

Listing 2: MakeEcho.b09

```
PROCEDURE Maketcho
(* Generates the binary module Echo *)
(* Level 1 - change all /dd to /d0 *)
DIM path,byt:BYTE
DIM count:INTEGER
CREATE #path,"/dd/cmds/Echo":WRITE
FOR count=1 TO 249
READ byt
PUT #path,byt
NEXT count
CLOSE #path
SHELL "attr /dd/cmds/Echo e pe"
END
DATA 135,205,0,249,0,13,17,129,209,0,38,2,147,69,99
DATA 104,239,4,40,99,41,49,57,56,56,32,83,46,66,46
DATA 71,111,108,100,98,101,114,103,15,0,15,1,49,68
DATA 52,32,230,128,193,92,38,96,230,128,193,92,39,90
DATA 193,36,38,6,141,108,141,71,32,80,202,32,193,99
DATA 39,82,193,102,38,4,198,12,32,66,193,110,38,4,198
DATA 10,32,58,193,114,38,48,15,3,141,25,141,37,52,4
DATA 230,128,193,44,38,60,141,13,53,2,141,73,167,160
DATA 12,1,90,38,249,32,179,230,128,193,36,38,3,141
DATA 47,57,141,77,230,104,57,13,2,39,161,57,141,67
DATA 230,164,141,245,231,160,12,1,193,13,38,146,16
DATA 158,0,53,16,134,1,16,63,140,37,1,95,16,63,6,95
DATA 15,2,141,0,166,132,128,48,129,9,35,14,166,132
DATA 132,223,128,55,129,15,34,15,129,10,37,11,52,2
DATA 134,16,61,235,224,48,1,12,2,57,48,31,95,15,2,141
DATA 2,141,0,166,132,128,48,129,9,34,17,52,2,134,10
DATA 61,235,224,137,0,38,6,231,164,48,1,12,2,57,196
DATA 130,167
```

INTERCOM
CONTINUED FROM PAGE 20

Scanlon, 2 Eagle Lane, Simsbury, 06070, (203) 657-8373

FLORIDA

☎ The Color Computer 3 Users Group, Tom Batchelder, 6042 Syrcle Ave., Milton, 32570, (904) 623-4405

GEORGIA

☎ Atlanta Computer Society, Inc., Alan R. Dages, 4290 Bells Ferry Road Suite 10639, Kennesaw, 30144, (404) 469-5111 voice, (404) 636-2991 modem

IDAHO

☎ Snake River Color Computer Club, Emil Franklin, 1750 Carmel Drive, Idaho Falls, 83403, (208) 522-0220

ILLINOIS

☎ Cook County Color Computer Club, Howard Luckey, 10 McCarthy Rd., Park Forest, 60466-2122, (708) 747-0117

☎ Motorola Micro Computer Club, Steve Adler, 1301 East Algonquin Rd., Shaumburg, 60196, (708) 576-3044

IOWA

☎ Metro Area Color Computer Club, Joe Cavallaro, 2425 Ave A, Co. Bluffs, 51501, (712) 322-2438

☎ Mid Iowa & Country CoCo, Terry Simons, 1328 48th Street, Des Moines, 50311, (515) 279-2576

KENTUCKY

☎ Hardin County Color Computer Club, Paul Urbahns, 2887 Republic Ave., Radcliff, 40160, (502) 351-4757

LOUISIANA

☎ The CoCo SIG, Christopher Mayeux, 20 Gibbs Drive, Chalmette, 70043, (504) 277-6880 voice, (504) 277-5135 modem

MARYLAND

☎ Arkade, John M. Beck, 3513 Terrace Drive #D, Suitland, 20746, (301) 423-8418

MASSACHUSETTS

☎ NorthEast CoCo Club, Jose Joubert, 440 North Ave., Bldg. 9 #210, Haverhill, 01830, (508) 521-0164

MICHIGAN

☎ Color Computer Owners Group, Bernard A. Patton, 388 Emmons Blvd., Wyandote, 48192, (313) 283-2474

☎ Greater Lansing Color Computer Users Group, E. Dale Knepper, P.O. Box 14114, Lansing, 48901, (517) 626-6917

MISSISSIPPI

☎ Mississippi OS-9 User Group, Boisy G. Pitre, Southern Station, Box 8455, Hattiesburg, 39406-8455, (601) 266-2807

MISSOURI

☎ CoCoNuts User Group, Clyde Lloyd, 2116 N. Columbia, Springfield, 65803, (417) 866-8738

☎ KC CoCo, Gay Crawford, P.O. Box 520084, Independence, 64052, (913) 764-9413

NEBRASKA

☎ Bruce Gerst c/o Metro Area CoCo Club, P.O. Box 3422, Omaha, 68103

NEW YORK

☎ Erie County Color Computer Club, John A. Lombardo, 57 Chapel Ave., Cheektowaga, 14225

NORTH CAROLINA

☎ Raleigh CoCo Club, P.O. Box 10632, Raleigh, 27605, (919) 878-3865

☎ The Tandy Color Computer Users of Charlotte, Eric Stringer, 1022 Noles Dr., Mt. Holly, 28120

OHIO

☎ The Greater Toledo Color Computer Club, Bill Espen, 1319 North St., Bowling Green, 43402, (419) 471-9444

☎ Tri-County Computer Users Group, Ron Potter, 10914 Oliver Road, Cleveland, 44111, (216) 476-2687

PENNSYLVANIA

☎ Cumberland Valley Users Group, Thomas Martin, 9085 Newburg Road, Newburg, 17240, (717) 423-5525

RHODE ISLAND

☎ New England "CoCoNuts" Color Computer Club, Arthur J. Mendonca, P.O. Box 28106 North Station, Providence, 02908, (401) 272-5096 (Sig3)

SOUTH CAROLINA

☎ Spartanburg CoCo Club, Jesse W. Parris, 152 Bon Air Ave., Spartanburg, 29303, (803) 573-9881

SOUTH DAKOTA

☎ Empire Area Color Computer Users Group of South Dakota, Carl Holt, P.O. Box 395, Brandon, 57005, (605) 582-3862

TEXAS

☎ The Codis CoCo Symphony, William C. Garretson, 2902 Harvard St., Irving, 75062, (214) 570-0823

UTAH

☎ Salt City CoCo Club, L. Todd Knudsen, 6357 S. Lotus Way, West Jordan, 84084, (801) 968-8668

WASHINGTON

☎ Bellingham OS-9 Users Group, Rodger Alexander, 3404 Illinois Lane, Bellingham, 98226, (206) 734-5806

☎ Port O' CoCo, Donald Zimmerman, 3046 Banner Rd. SE, Port Orchard, 98366-8810, (206) 871-6535

AUSTRALIA

☎ Australian National OS-9 Users Group, Gordon Bentzen, C/- 8 Odin Street, Sunnybank, Queensland, 4109, (07) 344-3881

☎ Brisbane Southwest Colour Computer Users Group, Bob Devries, 21 Virgo St., Inala, Queensland, 4077, (07) 372-7816

CANADA

☎ Club d'Oridinateur Couleur du Quebec Inc., 8000 Metropolitain est, Anjou, Quebec, H1K 1A1, (514) 354-4941

GERMANY

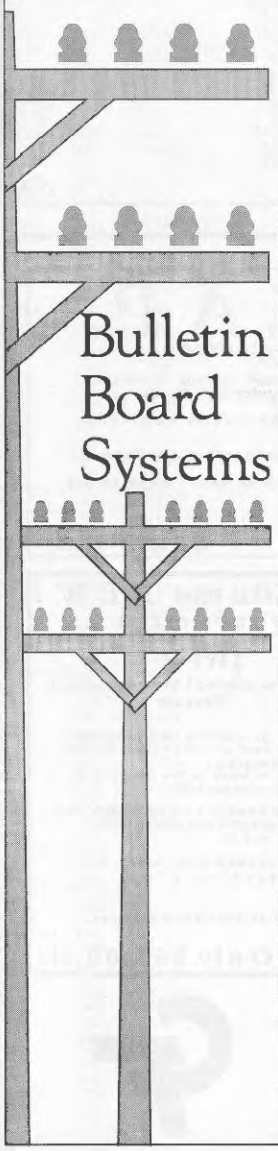
☎ OS-9 Users Group in Europe, Burghard Kinzel, Leipziger Ring 22A, 5042 ERFTSTADT, +49-2235-41069, (OS-9/6809)

THE NETHERLANDS

☎ European OS-9 User Group, Peter Tutelaers, Strijperstraat 50A, 5595 GD Leende, s88405777@hsepml.hse.nl, +31-4906-1971, (OSK)

PUERTO RICO

☎ Puerto Rico Color Computer Club, Luis R. Martinez, P.O. Box 2072, Guaynabo, 00657-7004, (809) 799-8217 or (809) 728-2314



BBS's

State/City	BBS Name	Access Number (Speed-Parity-Word Bits-Stop Bits)	Parameters	SysOp
Arkansas				
Sheridan	The Grant County BBS	(501) 942-4047	300/1200/2400-N-8-1	Eddie Gilmore
California				
Hollywood	Zog's Cavern BBS	(213) 461-7948	300/1200/2400-N-8-1	Alan Sheltra
Connecticut				
Manchester	Silk City BBS	(203) 649-9057	300/1200/2400-N-8-1	Darren Kindberg
Waterbury	Applause BBS	(203) 754-9598	300/1200/2400-N-8-1	Carmen Izzi, Jr.
Hawaii				
Ft. Shafter	CoCo'Nuts BBS Service	(808) 845-7054	300/1200/2400-N-8-1	Tommie Taylor
Idaho				
Idaho Falls	Snake River Computer Club BBS[1]	(208) 523-3796	300/1200-N-8-1	Jon Gould
Illinois				
Elmhurst	Glenside's Cup of CoCo BBS	(708) 428-0436	300/1200/2400-N-8-1	Tony Podraza
Lake Villa	Glenside CoCoRama	(708) 587-9837	300/1200/2400-N-8-1	Dave Barnes
Kentucky				
Elkhorn City	Cross-N-Crown BBS	(606) 754-9420	300/2400-N-8-1	Tim McIntosh
Michigan				
Manistee	Crystal Palace	(616) 723-0146	1200/2400-N-8-1	Nelson Howard
Taylor	J & L's CoCo Corner	(313) 292-4713	300/1200/2400-N-8-1	Jim Snider
Mississippi				
Hattiesburg	The OS-9 Zone[2]	(601) 266-2807	300/1200/2400-N-8-1	Boisy G. Pitre
New York				
Erie County CoCo Club		(716) 649-1368	300/1200/2400-N-8-1	Wayne Mullen
Wappingers Falls	The Dutchess CoCo	(914) 838-1261	300/1200/2400-N-8-1	Chris Serino
North Carolina				
Wilmington	Bill's Board	(919) 395-4366	300/1200/2400-N-8-1	Bill Medcalf
North Dakota				
Minot AFB	The 9-Line BBS	(701) 727-6826	300/1200-N-8-1	David Hensley
Ohio				
Columbus	Springwood BBS	(614) 228-7371	300/1200/2400-N-8-1	Edward Langenback
Pennsylvania				
Conshohocken	Charlie's Help Line	(215) 825-3226	300/1200-N-8-1 or N-7-1	Charles DiMartino
Rhode Island				
Central Falls	The Weather Connection II BBS	(401) 728-8709	300/1200/2400-N-8-1	Eric Chew
Virginia				
Fall Mills	Clem's Corner BBS[3]	(703) 322-4053	300/1200-N-8-1	Richard Douglas Bailey
Washington				
Fircrest	OS-9 Tacoma	(206) 566-8857	300/1200/2400-N-8-1	Chris Johnson
Wisconsin				
Marinette	Phoenix Interstate Data Systems[4]	(715) 732-1036	300/1200/2400/9600-N-8-1	Joe Boburka
Canada				
Twillingate, NF,	ColorNET BBS	(709) 884-2176	300-N-8-1	Jason Woodford
Windsor, Ontario	Color Connection	(519) 948-1879	300/1200-N-8-1	Cory Richert

Notes:
[1]Snake River Computer Club BBS supports all types of computers.
[2]The OS-9 Zone is up from 10 p.m. to 6 a.m. seven days a week.
[3]Clem's Corner BBS is up from 6 p.m. to 11 p.m. seven days a week.
[4]Phoenix Interstate Data Systems has a .75/hr charge for premium services, paid in advance.

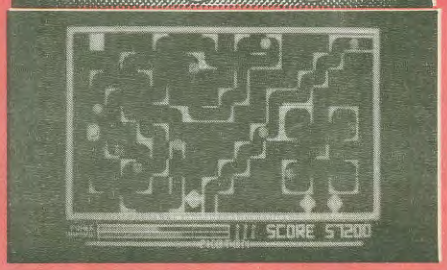
Advertisers Index

We appreciate your mentioning THE RAINBOW when you contact these advertisers.

SUNDOG SYSTEMS

PHOTON

NEW!



Energy is everything; your home world depends on it. However, someone or something is slowly siphoning it away. As your world's champion, you must climb into the experimental Power Tank to challenge this nemesis and his minions. Your key lies with the ability to teleport solid mass. Use this to manipulate and explore the endless stronghold of the enemy, and to exploit the free-floating DUPES (Dense Units of Photon Energy) to destroy the menacing Plasma Droids. Be cautious, though; those DUPES can be deadly, too! Photon, a fantastic new arcade game for your Coco3, contains spectacular 320x200 resolution, 16 color graphics, ultra-smooth 60 Hz animation, and loads of real-time music and sound effects. It will send your mind racing over endless possibilities, requiring quick decisions and reactions. Quite simply, Photon is incredibly addictive; it will deliver hours of excitement. Will you become your world's greatest hero, or just another energy slave? Req. 128K CoCo 3 and disk drive.

$34.95

GRAF 2.0 EXPRESS

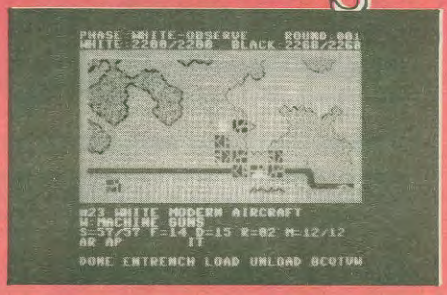
NEW!

GrafExpress 2.0 is a complete graphics and music programming environment. From the beginner to the accomplished professional, you can use GrafExpress to create lightning fast arcade games, graphic applications and utilities, and windowing multimedia demonstrations! The GrafExpress package includes two incredible systems. GrafExpress 16 works on all monitor types and offers support in 12 graphic resolutions (from 128x192 to 320x225). GrafExpress 256 offers 6 resolutions (from 128x192 to 160x225 on a composite monitor) in an astounding 256 colors! Ever see a CoCo do that before? Both systems include standard graphics commands (CIRCLE, FILL, etc.) that blow away the competition. For example, the BOX command peaks out at over 2 MegaPixels/second; that's 300 times faster than BASIC! 255 separate sprites of up to 100x100 pixels each are supported with window clipping and high-res pixel level collision checking. The 8-octave/4-voice music synthesizer has independent envelope, waveform, and volume controls, a 7+ KHz sampling rate, and much more. Other features include text/graphics mixing, different font sizes, fast window copying and scrolling, picture save/load, easy implementation from both BASIC and assembly language, multiple screen animation, and support for 128K/512K, double speed, and the high-res joystick interface. The package also contains support programs that are worth the purchase price of GrafExpress alone! These include an introductory demo, a picture editor, a waveform editor, and an art program that supports 256 colors! GrafExpress also comes with a 50 page manual that fully explains all of its incredible features. If you do any graphics programming or simply want to see what your little CoCo is capable of, GrafExpress is a must! Req. 128K CoCo 3 and disk drive.

$34.95

War Monger

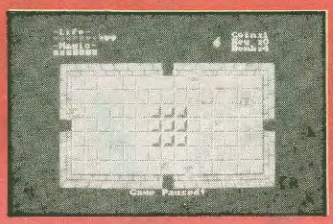
NEW!



The world is in unrest. Power-hungry villains and evil warlords are readying their forces. It falls to you to lead your people against these armies, and only your best strategic plans can save the day. Fight the good fight in any era or locale. Play a simple game of capture the flag armed with water balloons, or climb into the cockpit of a 100 foot high armored warrier. Explore the deepest dungeons, defend your galaxy, or create your own scenarios with this incredible war game construction set/simulator. Your imagination is your only limit. You will deploy your forces with total control over hostile terrain while you scroll a graphic bird's-eye window over an immense world. War Monger has terrific 320x200 resolution, 16 color graphics and includes a tile editor to create or edit your own. Play against the computer, battle with another player, or simply watch the computer plot against itself. The enemy is everywhere. Are you ready to take on the challenge as the War Monger? Req. 128K CoCo 3 and disk drive.

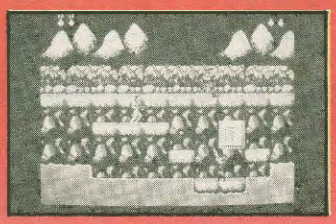
$29.95

The Quest for THELDA



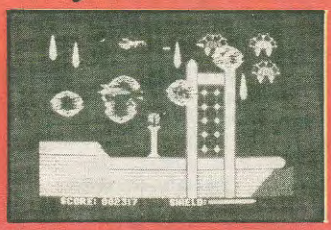
An immensely popular 128K CoCo 3 arcade/adventure. Over 500 screens of fast fantasy action and puzzle solving. Great graphics and sound effects. **$34.95.** Hint book only $4.95.

★ THE ★ CONTRAS



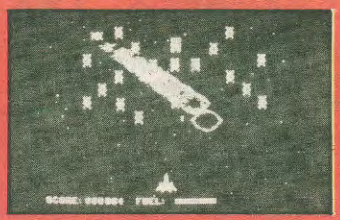
A 512K two player futuristic combat arcade game. Full screen 320x225 hardware scrolling and smooth animation. Back-ground music score and sound effects! 512K Coco3 only. **34.95.** Shipping soon!

Crystal City



This was THE game of '91! Ultra-fast space action with hardware scrolling on a 128K CoCo 3. Wild sound effects and over 30 MegaBytes of amazing graphics! **34.95.**

Zenix

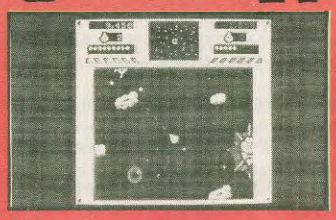


Lightning fast arcade game for the 128K CoCo 3. Terrific 320x225 graphics, back-ground music score and sound effects, and out-of-sight game play. **$29.95.**

ALSO AVAILABLE

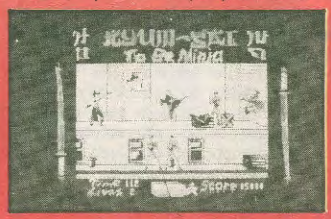
Warrior King	
CoCo 3	$29.95
In Quest of the Star Lord	
CoCo 3	$34.95
Hint Sheet	$ 3.95
Hall of the King 1, 2 or 3	
CoCo 1 - 3	$29.95 ea.
Hall of the King Trilogy	$74.95
White Fire of Eternity	
CoCo 1 - 3	$19.95
Dragon Blade	
CoCo 1 - 3	$19.95
Champion	
CoCo 1 - 3	$19.95
Paladin's Legacy	
CoCo 1 - 3	$24.95

Sinistaar



Everyone loves this 512K arcade game. 3 disks packed with spectacular graphics and eerie background digital sound effects. 512K CoCo 3 only. **$34.95.**

KYUM-SAI TO BE NINJA



The best selling 128K CoCo 3 martial arts arcade game. Now available in both RS-DOS and OS-9 versions. Play the incredible combat experience you've been missing under the operating system of your choice! **$29.95.**

SOUNDRAM

A polyphonic digital sound sequencing system for your 128K/512K CoCo 3 with a user-friendly point-and-click graphic editor. Create music scores with your own sounds or from the many we provide. **$34.95.**
Sample instrument disks: 6 sides of sampled sounds/instruments. Only $12.95 each or $29.95 for all three.

Visa, Mastercard, Check, Money Order, and COD (USA only, please) accepted. All foreign orders must be sent in US currency Money Orders. Include $2.50 for shipping in USA and canada. $5.00 Foresign. $3.00 extra for COD orders. PA residents add 6% sales tax. Dealer inquiries welcome. Authors, we're looking for new software!

Sundog systems

SUNDOG SYSTEMS

P.O. Box 766 • Manassas, VA 22111

703/330-8989