

Feature Article

## DS-69B VIDEO DIGITIZER

by STEVE RICKETTS

When I bought a DS-69A video digitizer from MicroWorks five years ago, I was quite happy with the method used to control the brightness of the image. This was done through software, and there were no knobs I had to fiddle with. Then came the CoCo 3 and the DS-69B upgrade, which also uses software control for image brightness. There are times, however, when it sure would be nice to have a brightness setting somewhere between those supported by the software.

The need for finer level control is especially strong now that it's possible to create beautiful 4096-color (simulated) pictures with the DS-69B. Such images are created by using three individual colored screens: red, green and blue. The brightness levels of the blue and green scans are critical for an accurate rendition of the colors contained in the image you're digitizing, so I had to find a way to overcome the software control. (Those who are interested in creating simulated 4096-color images with the DS-69B and a videocamera are welcome to contact me for more information on the process.)

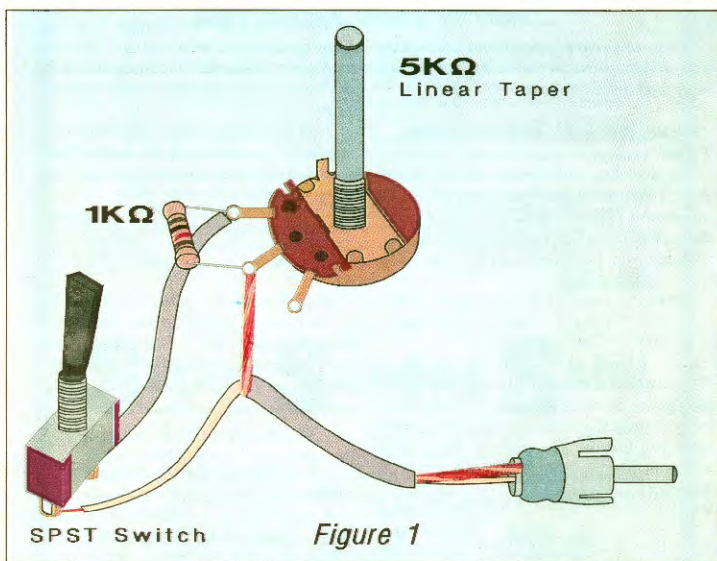
At the suggestion of Richard Trasborg

and Michael Trammell, I decided to build a simple circuit to regulate the signal before it enters the digitizer, instead of relying on the software to handle it after the fact. There are many ways such a circuit could be constructed (some simpler, some more complicated), so I'll just explain what I did to build mine. Fortunately I was able to construct this little project using only parts from my junk box. But even if you use all new parts from your local Radio Shack, the project should cost no more than \$10. In addition, while I use the level control with the DS-69B, it works well with older models of the DS-69 and earlier CoCos.

### Under Construction

The basic concept of the level control is to short the positive and negative sides of the video input signal, using varying amounts of resistance. To get the variable resistance, I used a 5K linear-taper potentiometer. The circuit diagram for the level control is shown in Figure 1. All the parts needed are listed in Figure 2. Except for the Y cable, all these parts are mounted in a mini project box.

SEEDS-69B ON PAGE 12



## SNOWFLAKE

### Screen Saver

Feature Program

by Bill Budenholzer

Snow flakes are crystals that are formed when water molecules freeze and melt as they rise and fall in the clouds. They form in apparently random six-sided shapes, then fall to the earth.

Flake Maker is a BASIC09 program that draws random six-sided shapes, simulating the appearance of snow flakes. Actually, a couple of different procedures are used to make the Flake Maker system work. These programs use the graphics capabilities of OS-9 Level II on the CoCo 3.

Listing 1 contains the two main procedures forming the Flake Maker system. The first of these, Flake\_Maker, allows the user to select the background color. It also repeatedly runs the second procedure, Flake, which generates the individual snowflakes. The procedure

SEE SNOWFLAKE ON PAGE 11

### In this issue:

OSK Experiences, Part 2	by John Donaldson	15
Received and Certified		18
18	❖ Snowflake Screen Saver	
	by Bill Budenholzer	1
14	☞ Three Points of Chaos	
	by Tom Beshara	6
8	Tips, Tricks and Traps	
	by Tim Kientzle	3
4	☞ What's in a Byte?	
	by Keiran Kenny	16
1	Product Reviews:	
6	Life and Death	
	from BDS Software	9
10	TasCOM	
	from CoCoPRO!	13
2	Letters to Rainbow	



**THE RAINBOW**

**Editor and Publisher**  
Lawrence C. Falk

**Managing Editor** Cray Augsburg

**Associate Editor** Sue Fomy

**Submissions/Reviews Editor**  
Julie Hutchinson

**Technical Editor** Greg Law

**Technical Assistants** Ed Eilers

**Contributing Editors** Tim Kientzle,

Martin Goodman, M.D., Eddie Kuns

**Art Director** Debbie 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 Clavley

**Director of Creative Services**

O'Neil Arnold

**Chief Bookkeeper/Dealer Accounts**

Beverly Quick

**Asst. Gen. Manager for Administration**

Tim Wheilan

**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)220-4492

**For RAINBOW Advertising and Marketing Office Information, see Page 19.**

THE RAINBOW is published every month of the year by FALSOFT, Inc., The Falsoft Building, 9509 U.S. Highway #2, P.O. Box 985, Prospect, KY 40059, phone (502) 228-4492. THE RAINBOW, RAINBOWest and THE RAINBOW and RAINBOWest logos are registered trademarks of FALSOFT, Inc. Second class postage paid Prospect, KY, and additional offices. USPS N. 705-050 (ISSN No. 0745-4737). POSTMASTER: Send address changes to THE RAINBOW, P.O. Box 985, Prospect, KY 40059. Authorized as second class postage paid from Hamilton, Ontario by Canada Post, Ottawa, Ontario, Canada (GST No. R125-334092). 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-ended 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. • Subscribers 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. \$58. Air mail U.S. \$102. 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 10/12th the subscription amount after two issues are mailed. No refund after mailing of three or more magazines.



**LETTERS**

**NX-1000 Conversions and 512K**

**Editor:**

I am trying to convert *Color Graphic Banner* (May 1984, Page 41) for my Star NX-1000 Color printer but cannot get it to work properly. The program was originally designed to run on the CGP-220. When I change the control codes to those for my printer, all it does is print small letters in the color I picked. I would greatly appreciate any help.

I would also like to know where I can get replacement chips for the CoCo 3 512K upgrade. I have the Tandy upgrade, but Radio Shack's chips are priced too high for my budget.

David Sekirka  
1687 N. Michigan, Lot 106  
Plymouth, IN 46563

*The CGP-220 uses such a unique set of control codes that it would be a major undertaking to alter the program for use with another printer. On the other hand, perhaps another reader has already done this or would be willing to give it a shot. Also, consider other banner printing programs; Zebra offers one, as does Sub-Etha, whose products are available through CoCoPRO!*

**Access to 512K?**

**Editor:**

I am writing a new program for the CoCo 3, and I need to gain access to the computer's full complement of 128K (or even 512K). I want to do this via machine language, but I need some help. I would appreciate hearing from others who have such experience and knowledge.

David Polonsky  
4 Tudor Court, Apt. 15  
800 North Broad Street  
Elizabeth, NJ 07208

**Wants to Upgrade Up North**

**Editor:**

I am taking a college course in electronics and want to use my CoCo as a word processor (and for any other applications I can). The problem is that I live in a small town in northern Ontario, and it is hard to find anything for the CoCo up here.

I have a Tandy Color Computer 3 and an FD-501 disk drive. I am looking for a 512K (or bigger) upgrade kit, an extra disk drive, a printer and any available software. I have checked at the Radio Shack stores in Canada, but the employees there seem to be interested only in selling new computers, not in supporting the CoCo. I have enjoyed using my CoCo and feel it is "big enough" for what I need.

Robert Hynds  
P.O. Box 1307  
Hartleybury, ON P0J 1K0  
Canada

*Many of the products you mention are available through advertisers in THE RAINBOW. Owl-Ware offers disk drives and other hardware items, and CoCoPRO! carries Simply Better, which to the best of our knowledge is the only currently available Disk BASIC word processor.*

**The DMP-105 and Color File II**

**Editor:**

First, thank you for an excellent magazine—I plan to renew my subscription. The support available through THE RAINBOW is incredible, and things such as the 6309 chip paint a bright future for the beloved CoCo.

I am having a problem with the *Color File II* Program Pak. I am able to save and load files to cassette but am unable to print the files on my DMP-105 printer (which is set at 2400 baud) even though I set the print options to match. Any help would be greatly appreciated.

Also, in past issues of THE RAINBOW I have seen programs that transfer ROM Paks to disk, but I haven't seen any such programs lately. I am unable to check on Delphi or other BBSs since I don't have a modem.

Thomas F. Williams  
Quarters 2944-D  
Quantico, VA 22134

*Item 7 on the Printer Options menu of Color File II controls the baud rate at which data is sent to the printer. The default value is 88 for 600 baud. To tell the computer to send at 2400 baud, change this value to 18.*

*While there have been a few ROM-Pak transfer programs in the past, we haven't seen many new arrivals in this area. Most of the information being passed around at this point deals with breaking the copy protection schemes used on later games marketed by Tandy—a procedure we don't support. Further, according to the software license granted to purchasers of ROM Paks, transferring the programs to different media (e.g., disk) is illegal.*

**Needs a Disk Drive**

**Editor:**

Am I in a bottomless pit? I have a 512K CoCo 3 with a cassette recorder. This is a last-chance attempt to find a disk drive for my system. I could use any information or addresses that might help me find one. Names of companies that might have one on a back shelf would also be much appreciated. Thank you.

James Kirby, Jr.  
P.O. Box 387  
Elgin, OR 97827

*Does any one happen to have an extra disk-drive system they'd be willing to part with to help a desperate CoCo user?*

**Easy storage for Peeks and Pokes**

**Editor:**

In rereading older issues of THE RAINBOW, I see a lot of great peeks and pokes. To store them, I enter them as BASIC lines, using the addresses for line numbers. For example, following are two lines from my pokes file:

```
65495 * Poke.0: High Speed - P1A
65497 * Poke.0: Double High-Speed SAM - blows CC2 VDG
```

This method automatically puts the peeks and pokes in numeric order, making them easy to find. If you choose to do this, make sure you use the REM marker. Also, don't renumber the program with RENUM, or you'll be sorry!

Charles Scanlon  
2 Eagle Lane  
Simsbury, CT 06070-1703

**Reviewing Reviews**

**Editor:**

I am very pleased with the review of my product, *The CoCo Collection*. However, I want to let you know the description of the *Lock* program was not clear. The reviewer was correct in writing that it is a "security program," but he failed to say that *Lock* is

intended for file security rather than computer security. The program is designed so that important software could be locked and executed or unlocked only by the person who enters the correct password.

I would also like to note that the program *DOS Commands* has been updated. Until now the program displayed the directory only until it reached a killed file. Now it displays the entire directory.

In addition, I recently added a new game, *Scramble*, to the collection. *Scramble* supports up to nine players. Each player is given 10 random letters from which he must build the largest possible word. A number of people have tried it, and all seemed to like it.

Finally, I am working on a couple of other programs, including one I call *Typing Teacher*. As you might have guessed, it is a program designed to teach novices how to type. A lot of work went into creating effective typing exercises, and the program has been proven to work.

Sheldon Parsons  
P. O. Box 117  
Beaumont  
N.D. Bay, NF A0J 1A0  
Canada

**Looking for GMX**

**Editor:**

I am having considerable trouble finding an address for GMX (formerly Gimix Systems). That company produced a micro-mainframe called the 20/20, and I would like to contact them. Do you by any chance have this address?

Ray Hightower  
Price Daniel Unit  
5938 F.M. 1673  
Snyder, TX 79549

*We sure do. You can contact GMX at 3223 Arnold Lane, Northbrook, IL 60062-2406, (708) 559-0909.*

**D.L. Logo on the CoCo 3**

**Editor:**

Recently while rummaging through close-out software at a Radio Shack store, I found and purchased a copy of *D.L. Logo*. When I attempted to run it on my CoCo 3, all I got was a garbled screen. Can *D.L. Logo* be run on a CoCo 3, or did I not load it properly? If the disk is damaged, is there anywhere I can obtain a replacement or copy?

I am also having a lot of difficulty using my DCM-6 modem, and the Radio Shack folks haven't been able to help at all. Any suggestions?

Ron Hyz  
5113-13th Avenue S.  
Gulfport, FL 33707

*The OS-9 Kernel on the D.L. Logo disk is for Level 1, which cannot be directly booted on the CoCo 3. The easiest way to use D.L. Logo on a CoCo 3 is to first boot OS-9 Level II, then insert the D.L. Logo disk and proceed as normal.*

**Looking for OS-9 Languages**

**Editor:**

I learned that, in addition to BASIC99, OS-9 supports Pascal, assembly language and C. However, I have seen no ads in THE RAINBOW from companies offering these compilers. Can you tell me if they are available and where I can get them?

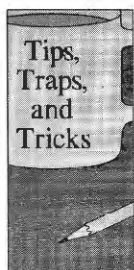
B.N. Gregoire  
2006 de Cambrai  
St-Bruno de Montarville, PQ J3V 2J3  
Canada

*The OS-9 compilers for Pascal and C*

are sold as separate items and were originally marketed by Tandy. The OS-9 Level II assembler is included with Tandy's Development System (Cat. No. 26-3032). These products may still be available through Tandy's Express Order System (1-800-321-3133). Alternatively, CoCoPRO! often carries these products.

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.



TIM KIENTZLE

## Preventing Namespace Pollution

Almost everything in a computer program has a name; variables, functions and macros all need to have names associated with them. As you begin to tackle larger programs, it becomes increasingly difficult to keep track of what each name means and which names have already been used. On the average, my assembly-language programs use about one name for every two to four lines of source code, so a 5,000-line assembly-language program will have around two thousand names in it. The local scoping rules in C help some, but organizing names is still a serious problem. And BASIC, with its two-character limit on variable names, makes it very difficult to keep track of names.

The collection of names or labels used in a program is sometimes called the *namespace*, and I find the term *namespace pollution* a particularly vivid metaphor for a problem that commonly arises in managing larger programs. What we'd like to consider are some techniques and approaches that can be used to reduce the pollution levels.

### Names Have Structure

Just as larger programs should be broken into smaller pieces in order to make them more manageable, the same applies to names. By imposing some structured approach on the way we create names, we can make our lives much simpler. Not only does it become easier to create new names, the names are also more meaningful.

Many programmers break names into three parts: a prefix, a root and a suffix. Usually the prefix and suffix identify what part of the program the name belongs to (such as *L* for a library function) or what kind of object it is (e.g., *I* for an integer). The root is created by the programmer to describe the use or purpose of the object. For temporary variables, the root is sometimes empty or is simply a digit (e.g., *i1* and *i2* for temporary integer variables). How the three parts of a structured name are separated is largely a matter of personal style. Sometimes case can be used, as in *iSize* for an integer variable that represents a size. Sometimes a punctuation mark serves the purpose, as in *I\_SIZE*.

In developing an approach to naming, you need to consider what specific prefixes and suffixes will mean. Also think about which root names you'll use most often (e.g., *Max*, *Min*, *Size*, *Limit*, *Count*, etc.), and how these will be combined to form complete names.

### Naming by Location

One way to use this idea is to mark each name by the part of the program it belongs to. I have found this especially useful for generating labels in assembly-language source code. Within a routine named *mul16*, I might use the labels *mul16a*, *mul16b* and so on. Here, the name of the subroutine becomes the prefix. Length considerations

sometimes make it impossible to use the full name, but the general idea can still be applied. In C, this is unnecessary for local variables but can be useful for structures and unions. Adding the structure or union name as a prefix or suffix can help you avoid a problem common with older compilers as well as provide an easy double-check on your own code.

Some programmers actually reuse large amounts of code from previous projects. In these cases, it can become necessary to use prefixes to identify not only the subroutine or component but also the program. This helps you avoid conflicts when combining pieces from several different programs.

To prevent names from becoming excessively long, it's a good idea to use just one or two letters for each prefix. For example, a subroutine from the print portion of a database package might be named *dprFormat*, where the *d* indicates that this is part of the database program, and the *pr* indicates the function is part of the print routines. Extending this convention to file-names makes it even easier; the *dprFormat* function might be stored as *dprint.c* or even *dpr.c*. [Editor's Note: For an example of this approach in action, see the source code for Phil Scherer's BBS9 starting on Page 18 of the October 1992 issue.]

In BASIC, this technique is quite common. Often, programmers use the same first letter to start the names of all variables used in a specific subroutine. For example, all variables in the *Print* subroutine may start with *P*. If you need more than 36 variables, you may use two or more different prefix letters. (While only the first two characters are used by the CoCo for name differentiation, you can use more than two characters for BASIC variable names.)

### Naming by Type

Naming by place is frequently used for functions and labels, but another technique is often more useful for naming variables (especially local variables). In this approach, a prefix or suffix is used with the name to indicate the type of the variable it specifies. This approach is heavily used in FORTRAN, where variable names beginning with the letters *I* through *N* are assumed to represent integers, and other variables are assumed to be real valued. BASIC also uses this in its convention that variables with names ending in a dollar sign are assumed to be string variables. Gaining popularity among C programmers is a very sophisticated version of this approach in which a coded prefix is used to indicate the type of variables. When combined with the idea about naming structure elements mentioned above, you end up with names such as *iWidth\_rect* for a field of *struct rect* which holds the width of the rectangle as an integer. Although such names look cumbersome at first, they become quite useful as your programs become more complex.

### Some Examples

Techniques for keeping track of names have been in use for a long time, and it's worthwhile to consider several before trying to develop a style that suits you. Following are some common approaches:

☛ **OS-9 System Calls** — most of the names defined by OS-9 begin with a prefix consisting of one or two uppercase letters followed by a punctuation mark. Some examples include *I\$* for function system calls, *I\$* for I/O system calls, *SS.* for *setstat* codes, and *D.* for system direct-page variables.

☛ **Hungarian convention** — many C programmers have adopted a convention in which each variable is prefixed with a type

code. By using *i* for integer, *l* for long, etc., and augmenting the name with such things as *p* for pointers, a programmer should immediately be able to properly reference *iSize* as an integer indicating a size, or *pSize* as a pointer to a long integer indicating the correct size. (Easy question: What's the *type* of *\*pSize*?) By augmenting names with short prefixes for constructed types, this method makes it easier to create new names (a temporary *char \** might be called simply *pc*) and helps to discourage certain common programming mistakes (such as confusing a pointer-to-long with a pointer-to-int).

☛ **C-library conventions** — the ANSI standard defines some standard methods of identifying certain C library functions. Unfortunately most of the names in the C library were developed by a variety of people over a long period of time, so there is no comprehensive standard to the approach used in creating these names. Among other conventions, the ANSI standard dictates that programmers should never use names beginning with an underscore since those are reserved for use by the compiler and library. There are also families of functions whose names are prefixed with *is*, *to* or *str*.

☛ **FORTRAN conventions** — as mentioned above, there is a long-standing convention in FORTRAN that variables beginning with *I*, *J*, *K*, *L*, *M* or *N* are integer variables. All other variables are real. This has been used in naming the standard functions (e.g., *ABS* is the real absolute-value function; *IABS* is the integer version). This extends to the identification of double-precision functions with a leading *D* and complex functions with a leading *C*.

☛ **BASIC conventions** — the BASIC language has a convention that variable names ending with a dollar sign are string variables. In BASIC90, this is not enforced (any variable can be defined as a string variable), but the convention is widely used even there.

### Consistency is the Key

Whatever method of naming you decide to use, the most important thing is that you use that method consistently throughout a program. In the process of working with a specific method, you may well find problems with that method, but you should resist the temptation to change it until you've finished writing the program. Using different naming styles within the same program inevitably leads to a great deal of confusion and frustration.

*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.*

## SOFT SECTOR

The PC Compatible Magazine

Special Back Issue Offer —  
Magazines \$1 each  
Disks \$6 each —  
while supplies last!

The PC Compatible Magazine  
**SOFT SECTOR** March 1992  
**SysStat**  
Understanding Memory  
Printer Control  
Sorting Array Data  
Easy File Manipulations  
Prompts with style and more

The PC Compatible Magazine  
**SOFT SECTOR** July 1985  
**Patriotic Graphics**  
Maze Escape  
Stellar Blast  
Scrolling  
A Look at Memory  
Utility Commands  
Technical Q&A and more

To order, call (800) 847-0309  
9 a.m.-5 p.m. EST.

**BONUS OFFER:**  
Soft Sector Binders \$3 Each



With the CoCo 1 and 2, it is especially important to make sure the printer is online before sending data to it. Include the following line in your BASIC programs to determine whether or not the printer is ready:

```
A=(PEEK(65314) AND 1)
```

If the value returned in Variable *A* is 1, the printer is *not* ready. (You can use any numeric variable in place of *A*.)

Feature Program

# DELUSIONS OF BINGO

by Steve Ostrom

9	18	FREE SPACE	46	66
14	22	38	53	75
3	16	40	47	63



Recently while playing *Bingo* with my two young daughters, I had a great idea for using my Color Computer — I'd write a program to print out all the possible *Bingo*-card arrangements. I had never seen a program that does this, yet it seemed to me it should be easy to write one.

Instead of just charging in and writing the program (as I normally do), I decided to make the task a little more interesting. So I dusted off my old college statistics book. The first realization to hit me was that it had been over 20 years since my last statistics class. So I dusted off that part of my brain as well, then proceeded to rediscover the subject and apply it to the task at hand.

Each *Bingo* card has five columns, the first of which contains five vertically arranged numbers that may range from 1 to 15. The second column also contains five numbers, but these values range from 16 to 30. The third (middle) column holds only four numbers (because of the Free space in the middle) in the range 31 to 45. Column 4 contains five numbers that may range from 46 to 60, and the range for the five numbers in the fifth column is 61 to 75.

The older part of my brain (now more or less dust free) told me this arrangement sounded a lot like the subject of *probability*, specifically *combinations* and *permutations*. A quick check in my stats book indicated that *combinations* is a technique in which you are selecting *n* different objects taken *r* at a time with no attention given to the order of the objects. On the other hand, with *permutations* the order of the objects is important. Since the order of the numbers in a column is very important in *Bingo* (two cards containing the exact same numbers but in a different arrangement will not normally produce a "bingo" at the same time), I decided *permutations* was what I needed to use.

The example given in my statistics book dealt with determining the number of permutations of the letters A, B and C, taken two at a time. The possible arrangements are AB, BA, AC, CA, BC and CB, so the number of possible permutations is six. The general equation to figure this out, instead of writing down all the possible arrangements and counting them, is

$$nPr = n!/(n-r)!$$

Here, *nPr* does not stand for National Public Radio but is an abbreviated way of saying the Permutations of *n* different objects taken *r* at a time. The exclamation

point (!) means *factorial*. A factorial works sort of like raising a number to power. Simply subtract a value of one from the number, then multiply this by the original number. Now subtract one again and multiply the result by the previous product. This process continues until you reach zero. Following are a few easy examples:

$$\begin{aligned} 1! &= 1 \\ 2! &= 2 \times 1 = 2 \\ 3! &= 3 \times 2 \times 1 = 6 \\ 4! &= 4 \times 3 \times 2 \times 1 = 24 \end{aligned}$$

Let's try the equation and see if it really works for the example of the letters A, B and C above. There are three letters taken two at a time. Therefore

$$\begin{aligned} 3P2 &= 3!/(3-2)! \\ &= 3!/(1)! \\ &= 3!/1! \\ &= 6/1 = 6 \end{aligned}$$

It works! Now back to the *Bingo* card. In terms of permutations, Column 1 contains 15 numbers taken five at a time. Let's use the equation again:

$$\begin{aligned} 15P5 &= 15!/(15-5)! \\ &= 15!/10! \\ &= (15 \times 14 \times 13 \times 12 \times 11 \times 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1) \\ &\quad (10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1) \end{aligned}$$

We could multiply all these numbers and divide, or we could let the CoCo do it. But there is a way to make the job easier — we can cancel like terms in the numerator and denominator. After this, the equation reduces to

$$\begin{aligned} 15P5 &= 15!/10! \\ &= 15 \times 14 \times 13 \times 12 \times 11 \\ &= 360,360 \end{aligned}$$

It is interesting that both halves of this six-digit number are the same; but, of somewhat greater importance, it tells us there are more than a third of a million possible permutations of the first column alone! Since the second, fourth and fifth columns are also "15 numbers taken 5 at a time," each has 360,360 permutations as well.

Because of the Free space, the third column is 15 numbers taken four at a time. Therefore

$$\begin{aligned} 15P4 &= 15!/(15-4)! \\ &= 15!/11! \\ &= 15 \times 14 \times 13 \times 12 \\ &= 32,760 \end{aligned}$$

The total number of possible arrangements can now be calculated by multiplying the number of permutations for each column, or:

$$(360,360)^4 \times 32,760$$

which my CoCo tells me is 5.52446474 x 10<sup>26</sup> in scientific notation. To get a feel for how big this number is, imagine it as a 5 followed by 26 zeroes.

Well, now I know why I haven't seen a program to print out all the possible *Bingo* cards. I find it absolutely mind-boggling that such a simple game as *Bingo* could produce an almost unbelievable number of unique game cards.

Now what do I do with my great idea? I know ... I'll write an innocent-looking article on *Bingo*, sneaking in a statistics lesson on permutations. But what about the program I wanted to write in the first place? Obviously my original idea won't work — even at one printed card per second it would take billions and billions of years! So I'll settle for something a little less ambitious.

The relatively short program accompanying this article prints single pages with four *Bingo* cards each. Since the program uses no special printer codes, it should work with any printer. Each page can be played as a set of four cards, or you can split the page by cutting on the \*'s to make four single cards or two sets of two cards. The number areas on each card are the perfect size for using pennies or buttons as markers. Each printed card should be different, within the random-number capabilities of the CoCo.

The information provided in this article won't improve your odds of winning games at your local *Bingo* hall. Nor will it help you get rich playing the lottery. But the program is useful, and you might just have learned a little something along the way.

*Steve Ostrom has a bachelor's degree in mathematics and a master's degree in food science. He has been a CoCo owner and subscriber to THE RAINBOW since the beginning, and he enjoys programming the CoCo in BASIC and assembly language. He can be contacted at 12612 Cedar Lake Rd., Minnetonka, MN 55305-3944, (612) 546-7608. Please include an SASE when requesting a reply.*

```
16K Extended
The Listing: BINGO.CRD
1 'BINGO CARDS
2 'BY STEVE OSTROM
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
60 CLEAR#000
70 DIM$(A),N(4,4),L$(4,4),R$(4,4),M$(4)
80 CLS:PRINT#0,"BINGO CARD MAKER
":PRINT:PRINT"MAKE SURE YOUR PRINTER PRINTEHEAD IS PROPERLY SET AT THE TOP OF THE PAGE. THIS PROGRAM USES ALL 56 STANDARD LINES ON AN 8 1/2X11 SHEET OF PAPER, SO PROPER PAPER/PRINTEHEAD ALIGNMENT IS CRITICAL."
90 PRINT#0;0,"PRESS CENTER> TO PRINT...":INPUT A
100 CLS:PRINT#0;235,"WORKING..."
110 X=RND(-TIMER)
120 S$(1)=" 8888 I I I I N
N G G G 000 "
130 S$(2)=" 8 8 I NN
N G 0 "
140 S$(3)=" 8888 I N N
N G G G 0 0 "
150 S$(4)=" 8 8 I N N
N G 6 0 0 "
160 S$(5)="( 1-15) (16-30) (31-45) (46-60) (61-75)"
170 S$(6)=".....
180 S$(7)=".....
190 GOSUB 240
200 PRINT#-2,STRING$(79,"*")
210 PRINT#-2,STRING$(39,"*");""
220 GOSUB 240
230 GOTO 0
240 GOSUB 630
250 L$(2,2)="FREE":R$(2,2)="FREE"
260 PRINT#-2,S$(1);"";S$(1)
270 PRINT#-2,S$(2);"";S$(2)
280 PRINT#-2,S$(3);"";S$(3)
290 PRINT#-2,S$(4);"";S$(4)
300 PRINT#-2,S$(1);"";S$(1)
310 PRINT#-2,S$(5);"";S$(5)
320 FOR Y=0 TO 4
330 PRINT#-2,S$(6);"";S$(6)
340 PRINT#-2,S$(7);"";S$(7)
350 PRINT#-2,S$(7);"";S$(7)
360 FOR X=0 TO 4
370 M$(X)=L$(X,Y)
380 NEXT X
390 GOSUB 500
400 PRINT#-2,S$(0);"";""
410 FOR X=0 TO 4
420 M$(X)=R$(X,Y)
430 NEXT X
440 GOSUB 500
450 PRINT#-2,S$(0)
460 PRINT#-2,S$(7);"";S$(7)
470 NEXT Y
480 PRINT#-2,S$(6);"";S$(6)
490 RETURN
500 S$(0)=" *M$(0)* *M$(3)* *
M$(4)* *M$(2)* *M$(3)* *
*M$(4)* * * *M$(3)* *
510 FOR Y=0 TO 4
520 FOR X=0 TO 4
530 N(X,Y)=0:NEXT X,Y
540 FOR Y=0 TO 4
550 FOR X=0 TO 4
560 N(X,Y)=15*X+RND(15):FL=0
570 IF Y=0 THEN N610
580 FOR Z=0 TO Y-1
590 IF N(X,Z)+N(X,Y)=MID$(STR$(N(X,Y)),2,1)+"" THEN MID$(STR$(N(X,Y)),3,1)+""
670 NEXT X,Y
680 GOSUB 510
690 FOR Y=0 TO 4
700 FOR X=0 TO 4
710 IF N(X,Y) THEN N610
720 NEXT X,Y
730 RETURN
```

# BINGO

7	23	41	52	72
1	30	33	48	68

# SUNDOG SYSTEMS



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

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**

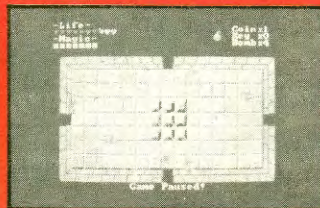
## War Monger



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 warrior. 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.

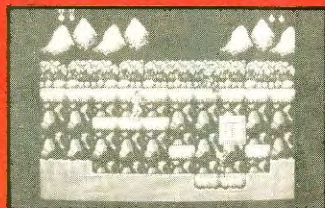
**\$29.95**

## The Quest for THELDA



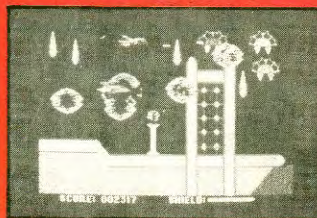
An immensely popular 128K CoCo 3 arcade/adventure. Over 500 screens of fast fantasy action and puzzle solving. Great graphics and sound effects. **\$34.95**. Hint book only \$4.95.

## ★ 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

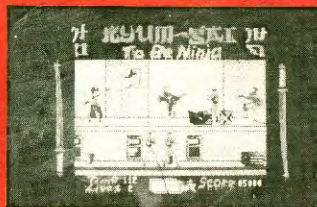
VISA, Mastercard, Check, Money Order, and CDD (USA only, please) accepted. All foreign orders must be sent in US currency. Money Orders include \$2.50 for shipping in USA and Canada. \$5.00 Foreign. \$3.00 extra for CDD orders. PA residents add 6% sales tax. Dealer inquiries welcome. Authors, we're looking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

## Sinistaar



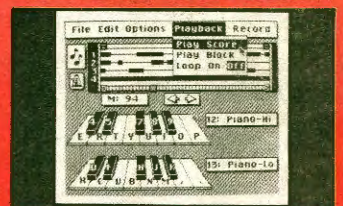
Everyone loves this 512K arcade game. 3 disks packed with spectacular graphics and eerie background digital sound effects. 512K CoCo 3 only. **\$34.95**.

## 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**.

## SOUNDTRAX



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.



# SUNDOG SYSTEMS



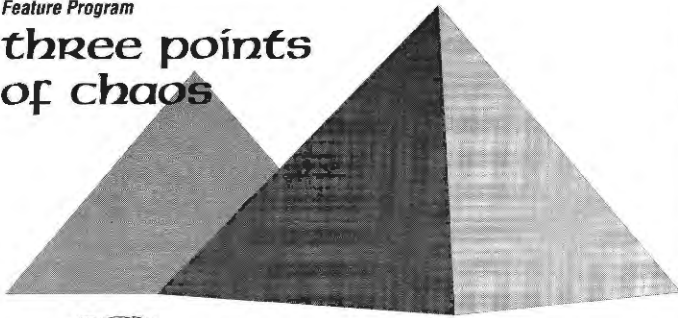
P.O. Box 766 • Manassas, VA 22111



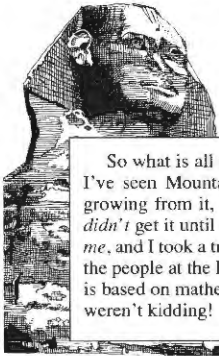
703/330-8989

Feature Program

# three points of chaos



by tom beshara



So what is all this "Chaos" stuff anyway? I've seen Mountains and a Ball with twigs growing from it, but I don't get it. Rather, I didn't get it until my curiosity got the best of me, and I took a trip to the local library. What the people at the library told me is that Chaos is based on mathematical formulas. And they weren't kidding!

I still don't fully understand, but I did find an interesting example of Chaotic structure even I could comprehend. Get a pencil and paper, a ruler and a game die. Draw the three points of a triangle and label them as follows:

Point	Label
Top	A(1,2)
Rt. Bottom	B(3,4)
Lt. Bottom	C(5,6)

Notice that the numbers used here correspond to the numbers on the six faces of the die. Pick a random point on the paper and label it R. Now roll the die and, whatever number is displayed, determine which of the three points it is associated with (A, B or C). Measure the distance between Point R and whichever point the die indicates, divide this distance in half, and place a dot at that point. This random point, halfway between R and the other point, becomes the new Point R. Now repeat the procedure.

If you keep repeating this procedure for, say, 8 to 12 hours, you'll see the finished product. I don't imagine many of you will want to actually do this, but if you at least do several repetitions over a period of a few minutes, you should get an idea of how this works. To make shorter work of the whole concept, I wrote a BASIC program to handle the task for me.

*Triangle* relies on the fact that the CoCo has the ability to display many different dots and to address those dots individually. It also has the ability to add and subtract numbers very quickly. To use the program, enter it and save it to tape or disk, then run it. You'll see an onscreen display of that horrendous task I asked you to perform only a few minutes ago.

Now let's use the CoCo manual to understand how the program works. First find the 64-by-32-pixel worksheet in the back of the Coco manual. Although *Triangle* doesn't use this exact screen (the program is designed to use the CoCo 3's HSCREEN2), the larger worksheet is much easier to read. Now, place a point at 30 dots across and 2 dots down, and label that point A(1,2). Place another dot at (10,20) and label that point B(3,4). Place the third dot at (50,20), labeling it C(5,6). Let's place the random dot R at (20,10).

The two parts of each location above represent the x and y coordinates, respectively, of our four points. For *Triangle's* purposes, the top point would be AX=30, AY=2, and the random point would be RX=20, RY=10. After the program places these points (again, we are performing manually exactly what the program does), it rolls an imaginary die. We'll say that the die roll is a 5, which indicates Point C.

To place the new random point R, the program determines if the current RX is greater than or less than CX. In this case, 20 is less than 50, so RX is less than CX. The new value for RX is determined as follows:

$$RX_{NEW} = ((CX - RX_{OLD})/2) + RX_{OLD}$$

$$= ((50 - 20)/2) + 20$$

$$= (30/2) + 20$$

$$= 35$$

Since the current value of RY is also less than CY, the same formula is used to determine the new value of RY. The result is 15, so we place a dot at location (35,15) and label it R. Now the CoCo rolls the die again, and the process continues.

The equation to use when RX is greater than the other point is as follows:

$$RX_{NEW} = RX_{OLD} - ((RX_{OLD} - CX)/2)$$

where CX might be replaced with AX or BX, depending on the die roll. Should one of these equations produce a fractional result (say, 32.5), the CoCo's INT function is used to truncate that decimal value.

Earlier I said that I still don't understand Chaos. But this little exercise has at least helped me understand some of the methodology behind it. I hope you find the subject of Chaos more enjoyable, too.

*Tom Beshara is a self-taught CoCo programmer. He has written inventory, sales and tracking programs for the children's clothing consignment store his wife Lorie owns, and is now working on a program to produce estimates at his auto body shop. Tom can be reached at 2633 Manassas Way, Colorado Springs, CO 80922. (719) 471-8888. Please include an SASE when requesting a reply.*

CoCo 3

The Listing: TRIANGLE

```
1 *TRIANGLE
2 *BY TOM BESHARA
3 *COPYRIGHT (C) 1992
4 *FALSOFT, INC.
5 *RAINBOW MAGAZINE
10 HCOLOR 0,14
20 HSCREEN2
30 F=20:P=176:R=300:T=176:L=160:
I=12:THE X,Y COORDINATES OF A. B
.C
40 HSET(F,P)'=POINT A
50 HSET(R,T)'=POINT B
60 HSET(L,I)'=POINT C
70 X=RND(300)
80 Y=RND(176)
90 HSET(X,Y)'= FIRST RANDOM POIN
T R
100 Q=RND(6)'=THE DICE
110 IF Q=1OR Q=2 THEN GOT0160'P
OINT A
120 IF Q=3OR Q=4 THEN GOT0170'P
OINT B
130 IF Q=5OR Q=6 THEN GOT0180'P
OINT C
```

```
140 GOT0100
150 *** CONVERTS THE X, Y COORD
INATES OF A, B, C TO A COMMON VA
RIABLE
160 D=F:S=P:GOT0190
170 D=R:S=T:GOT0190
180 D=L:S=I
190 IFX<D THEN XX=D-X:GOT0210
200 IFX>D THEN XX=X-D:GOT0230
210 XX=XX/2
220 X=X+XX:GOT0250
230 XX=XX/2
240 X=D+XX
250 Y=INT(X)
260 IFY<S THEN YY=S-Y:GOT0280
270 IFY>S THEN YY=Y-S:GOT0300
280 YY=YY/2
290 Y=Y+YY:GOT0320
300 YY=YY/2
310 Y=S+YY
320 Y=INT(Y)
330 HSET(X,Y)'= NEW MEASURING PO
INT R
340 GOT0100
```

Feature Program

# GET ON TARGET!

by Steve Puls

Have you got an itchy trigger finger? How about some target practice? If your answer to these questions is Yes, *Phaser Range* is here to help. *Phaser Range* is a short program that presents a moving target on the CoCo's screen. Using the joystick, you must aim at the target and pull the trigger (push the firebutton).

When you first run *Phaser Range*, you are asked to select a playing level from 1 to 20. Level 1 is very easy (perfect for those little CoCo nuts), and Level 20 is a real challenge. After you select the desired level, the game starts and a small circle, the target, starts moving around the screen. Use a joystick plugged into the right joystick port to aim the crosshairs (the little thing that looks like a plus sign). When you are on or near the target, push the firebutton and a circle forms on the screen. If the target is within this circle, you score a hit.

You have about a minute to hit a target; if you fail to hit it, the game restarts. After you've hit ten targets, the average number of shots it took you to hit each target is displayed. At this point, simply press the fire button to play another round. Press BREAK to exit.

*Phaser Range* runs on any CoCo with at least 16K and Extended BASIC. The high-speed poke is used in Line 10 to increase the playing speed. If you are using a CoCo 3, change this poke to POKE 65497,0. Also, since the computer is not slowed down when you press BREAK to exit the game, make sure you enter the normal-speed poke before performing any disk or printer I/O. CoCo 1 and 2 users should enter POKE 65494,0, and CoCo 3 users should enter POKE 65496,0.

*Phaser Range* is a safe way to develop and enhance hand/eye coordination. Best of all, it works year round and on rainy days, too.

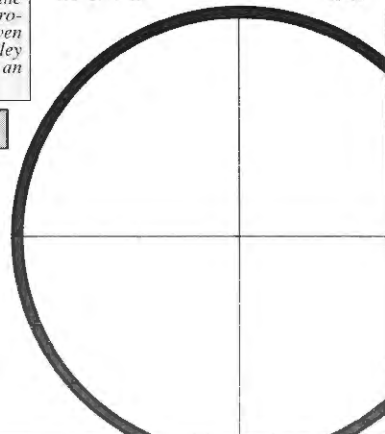
*Steven Puls is a high-school junior at Lyons Central School. Since his introduction to the CoCo six years ago, he has written a wide range of software for the computer. He aspires to a career in programming or a related profession. Steven can be contacted at 605 Pleasant Valley Rd., Lyons, NY 14489. Please include an SASE when requesting a reply.*

CoCo 3

The Listing: PHASER

```
1 *PHASER RANGE
2 *BY STEVEN PULS
3 *COPYRIGHT (C) 1992
4 *BY FALSOFT, INC.
5 *RAINBOW MAGAZINE
10 POKE65495,0
20 L=3000:RY=1
30 AS="NU5N05VL5NR5"
40 CLS0:INPJT "LEVEL (1 EASY - 2
0 HARD)":LV:IF LV<1 OR LV>20 THE
N 40 ELSE R=21-LV
50 PH00E4:SCREEN1,1:PCLS0
60 H=0:S=0
70 PG=0:FOR RD=1 TO 10
80 TH=RND(500)-L
```

```
90 PLAY"T2551255V3103CDEFGAR04CD
EFGB":H=0:V=0:X=160:Y=96:H1=0:V1
=0:V3=0:H3=0
100 H1=RND(320):V1=RND(190):H2=R
ND(50):V2=RND(100):T=RND(50)+50:
CIRCLE(H,V),3,1
110 IF H1+H2>329 OR V1+V2>189 TH
EN A=2 ELSE A=1
120 H=H1:V=V1
130 TIME=0
140 Z1=RND(15):Z2=RND(15):PSET(H
.V,1):X=JOYSTK(0)*4:Y=JOYSTK(1)*
3
150 ON A GOT0 250,270
160 IF H<0 OR H>255 OR V<0 OR V>
190 THEN GOT090
170 CIRCLE(H,V),3,1
180 IFPEEK(65280)=254ORPEEK(6528
0)=126THEN5=S+1:H3=160:V3=192:HIS
=X-160:V3=192-Y:V5=H3/20:S2=V3/2
0:FORAT=1TO10:PSET(H3,V3,1):PSET(
H3,V3,2):PSET(H3,V3,3):PSET(H3,V
3,0):H3=H3+1:V3=V3-S2:NEXTT:PLA
Y"V31L255T25505AC":CIRCLE(X,Y),R
.1:PCLS:GOT0220
190 DRAW"BM=X,-Y:C1"+AS:CIRCLE(
H,V),3,0:DRAW"BM=X,-Y:C0"+AS
200 PSET(X,Y,1):PSET(X,Y,0)
210 IF TIMER-T THENPLAY "03T25B
AGFEDCC02BAGFEDCCC":GOT0340 ELSE
GOT0 140
220 D1=H-X:D2=V-Y:IF (D1*D1)+(D2
*D2)<(R*R) THEN PLAY"T25503DCDC
DCDCDCDCDCDCDC":HT=HT+1:GOT0 240
ELSE A=RND(2):H=H+RND(5)*RND(2)-
1:V=V+RND(5)*RND(2)-1
230 GOT0 140
240 FOR CD=1 TO 20 STEP 5:POKE&H
FF22,RND(32)+223:CIRCLE(H1,V),CD,
1,3:PLAY "T251803C":CIRCLE(H,V)
,CD,0,3:POKE&HFF22,255:NEXT CD:
GOT0290
250 PSET(H,V,0):H=H+Z1:V=V+Z2:IF
H>H1+H2 OR V>V1+V2 THEN A=2:GOT
0160
260 GOT0 160
270 PSET(H,V,0):H=H-Z1:V=V-Z2:IF
H<H1-H2 OR V<V1-V2 THEN A=1:GOT
0 160
280 GOT0 160
290 NEXT RD
300 SCREEN 0,0:CLS0
310 PLAY"V31T251203CV2115CV10116
CV5164CCCV21255CCCCC":PRINT "YO
UR AVERAGE SHOTS PER HIT WAS "+
STR$(S/10)+" FOR ROUND":RN
320 PRINT @256,"PRESS FIRE BUTTO
N FOR NEXT ROUND":
330 IF PEEK(65280)<254 AND PEEK
(65280)>126 THEN 320
340 CLS3:PRINT@32,"PRESS FIRE BU
TTON TO RESTART"
350 IF PEEK(65280)<254 AND PEEK
(65280)>126 THEN 350
360 RN=RN+1:R=RLV/4:IF R<1 TH
EN R=1
370 GOT0 50
```





In our 10<sup>th</sup> 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 1496 based Systems with Socket for Weitek CoProcessor
- System and Video BIOS In Cache
- Large Tower Case : (93MHz, 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** Including One Full Year on Parts and Labor  
**on all systems!** 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®**

Pottstown Ave., RT. 663  
Pennsburg (215)-679-3389

**St. Onge Systems**

Wescosville  
Call for Appt. (215)-481-9775

**Computers & Games**

Muhlenberg Shopping Plaza  
Reading (215)-929-0540

# Proven Technology

On the Razor's Edge of the Color Computer Frontier

# 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.**

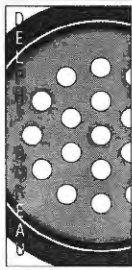
Now includes memory test, Ram Disk Lighting, Printer Lighting, and Back-up Lighting. All with an upgraded manual exclusive with OWL!

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



EDDIE KUNS

## Delphi Joins Internet

Delphi is now connected to Internet, a computer network between hundreds of thousands of computers worldwide. You can now send and receive electronic mail with people all around the world, including colleges and universities, CompuServe, America Online, MCI Mail and various businesses. You can also communicate with

people connected to Usenet and FIDO networks. If you want to use Internet, however, you must first register.

You can register for Internet mail by entering GO USING INTERNET from any Delphi prompt. This takes you to a menu where you have three options: You can exit if you change your mind, you can cancel if you have already registered and you later change your mind, or you can register. When you enter REGISTER, you first see several pages of information about the net-

work. You'll probably want to capture this information in a disk file for later reference. In any case, it is important that you read this information carefully. Finally, you are asked if you want to register. Make sure you understand the terms, limits and costs before answering Yes.

If you register to use Internet mail, you will be billed \$3 every month for Internet mail access. With this access comes a limit of 10 megabytes of Internet mail each month; this would be a tremendous amount of mail, so it is unlikely that you'll exceed this limit. Still, be careful. Also remember that mail is stored in your Delphi Workspace, and you pay for all disk space used beyond the first 50 blocks (25K).

On Internet are many computer mailing lists that function similarly to Forums on Delphi. There are mailing lists for every topic you can imagine — from fractals to computers to political discussions. If you join a mailing list, remember that you should read and delete mail frequently to avoid paying for more disk usage than you are willing. Some mailing lists have a large amount of mail daily; some have only a few messages per week.

One very nice feature of Internet is the ability to connect to Delphi from any other computer on the system, although you cannot connect to other computers from Delphi. On most Internet-connected computers, you can log onto Delphi by entering

TELNET DELPHI.COM

You should then see Delphi's USERNAME: prompt. Telnet carries no telecommunications surcharge, so you can connect to Delphi any time of the day using Telnet and pay only for connect time. Note that I am not referring to Telenet (now named SprintNet) which you reach via telephone; Telnet is a feature of Internet and is available only on computers directly connected to Internet. Most colleges have computers that are so connected. Ask your local system administrators if you are unsure.

Not all of Delphi's features are available when you connect via Telnet. For example, you cannot download any files. On the other hand, Telnet connections are almost always faster than 2400-bps modem connections.

If you want more information, call Delphi customer service at 1-800-695-4005 (617-491-3393 for people outside the United States).

*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 OS9 Online and can be reached online as EDDIEKUNS.*

## Advanced Forum Details

In the last two issues, I've explained in increasing detail the many features of Delphi's Forum areas. We'll continue this with explanations of the remaining commands.

If you read a Forum message and don't want to decide immediately whether or not to reply, use the TAG command to tag the message for viewing later. For example, suppose another user asks a question you can answer, but you don't know if somebody else has already replied. Simply enter TAG after reading the message, then continue reading Forum messages. Before you exit Forum, enter READ TAG to read the messages you tagged (when you exit Forum, the tags are removed from any messages you have tagged). Then you can decide whether or not to reply. It is polite to make sure how others have replied; otherwise several people may offer the same reply to a simple question. Note that once you read a tagged message, it loses its tag unless you tag it again.

If you want to keep a permanent copy of a Forum message, there are several options available to you. The quickest method is to capture the text to disk, assuming your terminal program supports that feature. Another option is the FILE command: if you have just read a Forum message, entering FILE SAVETHIS.TXT creates a file named SAVETHIS.TXT containing the Forum message in your Workspace. You can also add modifiers to the FILE command. For example, entering FILE SIG.TXT THREAD 57826 tells Delphi to save Message 57826 and all subsequent messages in its thread to a file in your Workspace named SIG.TXT. When you use this modifier, Delphi first asks if you are sure you want to save a range of messages into the file. There are many possible modifiers; to see the full list, enter FILE ? at the Forum> prompt.

Modifiers are also supported by other Forum commands. Two such commands are DIRECTORY (or DIR) and READ. In fact, these commands use most of the modifiers that work with the FILE command. Two of the more commonly used modifiers are FROM username and TO username. You can specify both in a single command, but only one FROM and one TO at a time. If you want to search a specific range of Forum messages, specify this by entering low-number:high-number. You can use either a colon (:) or a hyphen (-) to separate the two numbers. You can also specify open-ended ranges such as 12543- and -54312, which mean what you would expect — from Message 12543 to the last message, and from the first (or current) message to Message 54312, respectively.

Thus, if you want to search for all the Forum messages between 10000 and 11000 that were sent by or to Kevin Darling (KDARLING), you could enter:

```
dir 10000:11000 from kdarling or
to kdarling
```

If the resulting listing contains more lines than your current screen-length setting (/LENGTH), Delphi prompts you after each page. To get around this, you can specify NS (or NONSTOP) on the command line. Notice that I used OR in the above command line. If you do not specify OR or AND between consecutive modifiers, Delphi assumes you mean AND.

Another way to narrow a search is by using the date modifier. You can specify all messages BEFORE and/or SINCE specific dates to search a span of time rather than a range of message numbers. The date needs to be in a day-month-year format. For example, 12-Jan-1992 specifies January 12, 1992. As another example, to read all messages posted on March 28, 1992, without pausing between them, you could enter

```
read since 28-mar-1992 before
29-mar-1992 ns
```

You already know about READ NEW and READ WAITING. Remembering that READ and DIR share the same options, you can also type DIR WAITING to see which unread messages are addressed to you. Depending on how many messages have been addressed to you in Forum, this command may take a while.

You can also search through Forum messages by subject using the SUBJECT modifier. For example:

```
dir 57500-58000 subject "vm &
osk"
```

gives you a directory listing of all messages between 57500 and 58000 that have VM & OSK as their subject.

As with the FILE command, you can specify THREAD to follow a specific thread with the READ and DIR commands (but you may specify only a single message number). You can specify FT (short for follow thread) with READ NEW to see all new messages, following threads as they are encountered. Many people read messages by opening a capture buffer and entering READ NEW NS FT to read all new messages non-stop, following threads. Then they read the captured buffer offline.

Remember the IGNORE command I mentioned last month? This command can be used to ignore specific messages and threads, and it supports the same set of modifiers as FILE, READ and DIR. To see a full list of the modifiers supported by any of these commands, enter the command name followed by ?.

## Uploads at a Glance

In the OS9 Online Applications database, **Homer Meyer** (MEYEO01) released another set of utilities for creating and deleting directories. **John Semler** (SEMLER) released a new version of NIST for both OS-9/6809 and OSK. (Both versions are in the Applications database.) This program tells your modem to call the National Institute of Standards and Technology (NIST) telephone time service, and it sets your system time to the correct time.

**Hugo Bueno** (MRGOOD) uploaded a description of UUCPmap entries. Both this description and the USA UUCP site list contributed by **Rick Adams** (RICKADAMS) are useful to people who want to connect their CoCos to Usenet. **John Farrar** (TRIX) released the latest version of TRXMON, a program that functions like TSMON. TRXMON has many features, including utilities to handle password files with encrypted passwords.

In the Programmers Den database, **David Graham** (NIMITZ) and **Greg Law** (GREGL) uploaded OSK binary and source, respectively, for GNU Indent, a free C source-code formatter. In the OSK Applications database, **Mike Sweet** (DODGECOLT) released Version 0.70 of DOC. This is a demo version of his simple OSK word processor/editor. **Glen Hathaway** (COMPER) contributed a screen saver and a file-size finder, both ported from the CoCo. **Scott Kasten** (SKASTEN) released *Sphere*, a simple molecular modeling program.

In the CoCo SIG General Information database, **Alfredo T. Santos** (ALFRADIO) uploaded a list of the Tandy catalog numbers for all the items Tandy sold in support of the CoCo. **Robert Newhart** (BNEWHART) released in the Games database a simulation of the 1991 Iditarod race (from Anchorage to Nome, Alaska).

In the Music & Sound database, **Richard McNabb** (RICKMAC) uploaded ORCHMA, a public domain Orchestra-90-type music editor and compiler. **Terry M. Blackwell** (TMB) contributed TJUKEBOX, a program that sends *Musica II* files to your TV speaker, stereo cartridge, Orchestra-90 Pak or MIDI cartridge.

**Bernie Patton** (BERNIEP) announced the release of a patch to EDTASM that adds support for the new 6309 opcodes and many other new features.

## DATABASE REPORT

### OS-9 SIG

#### General Information

DATEDEX ANNOUNCEMENT  
EDELMIAR Ed Grestick  
HD6309 INFO  
JOHNBAER John Baer  
OS-9 MAJIC MAGAZINE  
ILLUSIONIST Michael Graftman  
PNW COCHREST REPORT  
SALZARD Rodger Alexander

#### Applications (6809)

INVENTORY PLUS VERSION 2.1  
MOHRT Jim Vestal & Tim Mohr

CROWN 9: TASK SCHEDULER  
WOAY Jim Martin  
PATCH FOR TSFMT  
BROADAXE Paul Brodnax  
MD AND RD FOR OS-9  
MEYEO01 Homer Meyer  
FUN LIB: AD-LIB STORY GENERATOR  
MOHRT Jim Vestal & Tim Mohr  
FONTASIZE: FONT DESIGNER  
JAMIEC Jamie Cho  
CCCLK: COGO CORNER CLOCK  
SAM35 Steve Mylonas  
NIST V1.1: TIME SERVICE DIALER  
SEMLER John Semler



<b>Telecom (6809)</b> UUCP MAP DESCRIPTION MRGOOD Hugo Bueno UUCP SITE LIST FOR U.S. RICKADAMS Rick Adams TRXMON - DEMO RELEASE TRIX John Farrar	<b>Tutorials &amp; Education</b> TEACHC: C TUTORIAL BOBKEMPER Robert Kemper
<b>System Modules (6809)</b> J&M PARALLEL PORT DRIVER WOAY Jim Martin INTFIX: INTERRUPT HARDWARE HACK DEANHOLDER Dean Holder	<b>CoCo SIG</b>  <b>General Information</b> TANDY SUPPORT FOR COCO ALFRADIO Alfredo T. Santos
<b>Music &amp; Sound</b> ST:TNG MONOLOGUE DEANHOLDER Dean Holder PORKY PIG SOUND FILE DEANHOLDER Dean Holder CORDS.LZH (STEREO) JOHNBAER John Baer SHORT CIRCUIT SAMPLES (MAC) MODEL299 Mark Steiner ST:TNG MINI THEME DEITERLEIN Chris Deiterlein	<b>CoCo 3 Graphics</b> MAC TO CM3 VERSION 2.1 RICKMAC Richard McNabb TORI AMOS 4096 DEANHOLDER Dean Holder FRANKENSTEIN DS69 DEANHOLDER Dean Holder REN & STIMPY DEANHOLDER Dean Holder TWO VIRGINS DEANHOLDER Dean Holder
<b>Programmers Den</b> HERSHEY FONT DATABASE DRDUDE Andy DePue GNU OSK INDENT NIMITZ David Graham GNU INDENT GREGL Greg Law	<b>Source for 6809 Assemblers</b> HD63B09E INFORMATION HYPERTECH Chet Simpson
<b>OSK Applications</b> DOC VERSION 0.70 DODGECOLT Mike Sweet DATADEX DEMO VERSION EDELMAR Ed Gresick UTIL1: TWO SIMPLE UTILITIES COMPER Glen Hathaway SPHERE MODELER SKASTEN Scott Kasten LHA V2.01 FOR OSK MIKEHAALAND Mike Haaland	<b>Hardware Hacking</b> HARDWARE INTERRUPT FIX DEANHOLDER Dean Holder
<b>OSK Telecom</b> XYDOWN FOR OSK TIMKIENZTLE Tim Kientzle	<b>Games</b> IDITAROD.ARC BNEWHART Robert Newhart THE DOCTOR JOCOCO Joseph Worthington SPIDER HYPE GAME JOCOCO Joseph Worthington
<b>OSK System Modules</b> UPGRADED MM/1 DRIVERS TIMKIENZTLE Tim Kientzle	<b>Class Graphics</b> DIGI'D PIC FOR COCO3/CLONES STEVEPDX Steve Ricketts
	<b>Music &amp; Sound</b> ORCHMA MUSIC EDITOR RICKMAC Richard McNabb SHAREWARE MUSICA PROGRAM RICKMAC Richard McNabb TJUKEBOX.ARC #2 TMB Terry M. Blackwell
	<b>Product Reviews &amp; Announcement</b> EDTASH6309 BERNIEP Bernie Patton

## Product Review

# Life and Death

The game of *Life* has been very popular with computer programmers and users since the beginning of the computer revolution. Perhaps the reason for this popularity is that *Life* demonstrates the computer's ability to maintain consistent random-pattern generation; besides, the changing screen looks pretty. Still, in the early days, all the game of *Life* could do was develop endless patterns of activity that were interesting to watch and difficult to understand. *Life and Death* from BDS Software, however, is truly a game.

In *Life and Death*, two players (or one player and the computer) compete in a life-and-death struggle. The *Life and Death* "world" is a 40-by-41 grid of squares, and the object of the game is to control more of the grid's 1640 squares than your opponent. To do this requires strategy and tactics — *Life and Death* is not an arcade or adventure game in any sense. There are no fancy graphics or sound effects.

One player's markers are red, and the other player plays blue. During a player's turn, he uses the arrow keys to move an onscreen cursor to the location where he wants to place a marker. I had a tough time determining where the tiny cursor was located: on an RGB monitor, the playing screen is a light green grid on a yellow background, and the cursor is a faint dash

— very hard for these old eyes to follow.

When each player has placed his markers, death and reproduction (a new generation) begin to occur, as with the original game of *Life*. This is the part you can't control and is the real mystery in the game of *Life and Death*. Believe me, the rules by which death and reproduction occur in *Life and Death* get pretty complex. I found it helpful to watch the demonstration game included on the disk. This proved to be an excellent way to understand how the game is played.

Since there is no limit to the possibilities, *Life and Death* doesn't name a winner. Rather, players must predetermine what "winning" is. Perhaps you'll play the game a specific number of turns, or you could impose a simple time limit. Another way is to specify a winning number of markers on the board (*Life and Death* displays this information for each player).

*Life and Death* is designed to run on any CoCo system with a disk drive. It works with televisions as well as composite and RGB monitors. The game is very easy to play, yet sometimes difficult to understand. On the other hand, so is *Life*. (BDS Software, P.O. Box 485, Glenview, IL 60025-0485, 708-998-1656; \$10 plus \$3 S/H.)

— Robert Gray

## GRANITE COMPUTER SYSTEMS ZOOM BRAND MODEMS NEW LOWER PRICES!

<b>14,400 BPS V.32bis/V.42/V.42bis modems. MNP1-5+LAPM.</b> Error Correction and Data Compression (much higher throughput - as much as 57,600 bps). Data only: External \$269 Internal \$249 (\$9 S&H*) With S/R FAX: 299 279
<b>9,600 BPS V.32/V.42/V.42bis modems. MNP1-5+LAPM.</b> Error Correction and Data Compression (much higher throughput - as much as 38,400 bps). Data only: External \$239 Internal \$219 (\$9 S&H*) With S/R FAX: 259 239
<b>9,600 BPS Send/Receive FAX Modems.</b> Send and Receive text/graphics files from/to your computer from/to Fax machines anywhere in the world. Full 2400 bps data mode. FAX & Data: External \$119 Internal \$109 (\$6 S&H*) With MNP1-5/LAPM (hardware): 139 129
<b>2,400 BPS Data Modems.</b> Data only: External \$75 Internal \$59 (\$6 S&H*) With MNP1-5/LAPM (software): 85 69

These are high quality modems made by Zoom Telephonics in the USA. Seven year factory warranty. Fully Hayes compatible. Terminal and FAX software is included with FAX modems. Terminal and Windows (PC) software available. MAC software also available. Cables available. \* S&H Canada: V.32/V.32bis modems \$13.00. Other modems \$9.00.

### GCS FILE TRANSFER UTILITIES - Version 3.0

GCS File Transfer Utilities provide a simple and quick method to transfer text/binary files from/to a variety of floppy disk formats.

Commands: PC,RS,FLEX disks: Dir, Dump, Read, Write  
PC Disks: Rename, Delete, Format

Handles most 5.25 and 3.5 formats. Any level sub-directories (PC). Binary files. Use pipes for multiple file transfers. Multi-Vue version can be used under Multi-Vue or as stand alone Shell commands.

Requires: OS-9 L2 for CoCo 3, L1 for CoCo 1 or 2; two drives (one can be hard/disk, one floppy 40 track DD DS); Multi-Vue for Multi-Vue version; SDISK3 for CoCo 3, SDISK for CoCo 1 or 2.

OSK version	\$99.95
Multi-Vue version	\$54.95 Standard version \$44.95
Version 3.0 updates (provide disk number)	\$25.00(OSK) \$15.00

D P Johnson Software SDISK or SDISK3 \$29.95 L1+L2 Utilities \$75.00

Shipping and handling - any software \$2.50 USA \$3.00 Canada

Orders must be prepaid or COD. COD is \$3.50 additional. VISA, MC, DISCOVER

571 Center Road, Hillsborough, NH 03244 USA  
(603) 464 - 3850

OS-9 is a trademark of Microware Systems Corporation and Motorola, Inc.  
MS-DOS is a trademark of Microsoft Corp. FLEX is a trademark of TSC, Inc.

The C Compiler for the CoCo has finally arrived...

## CoCo-C

CoCo-C is a complete RSDOS based C development package for the Color Computer not requiring the OS-9 Operating System. CoCo-C consists of five main programs: a Text Editor, a C Compiler, an Assembler, and a Library Linker which are all controlled by the CoCo-C Command Coordinator.

### Text Editor

A full featured screen oriented line editor for the CoCo3 developed by Bob van der Poel. Powerful editing and cursor commands with auto-indent and user defined macros make this a great editor for writing C or assembly language programs. A less sophisticated version for the CoCo 2 is also available.

### C Compiler

The CoCo-C Compiler is a full featured K&R style integer compiler specifically designed for RSDOS based systems. It has assembly language output, position independent code and can output ROM-able code if desired. Added features allow you to mix C, assembly language and BASIC commands within your program!

### Assembler

This symbolic assembler is capable of assembling files as large as available disk space. It supports a Motorola style syntax and outputs standard binary files ready for LOADM and EXEC. Options include list file output and generation of symbol table file.

### Library/Linker

The Library Linker is a utility which links the CoCo-C's 90+ function library with your compiled binary file, creating a stand alone executable ML file.

### Command Coordinator

The Command Coordinator is CoCo-C's main program. Its user friendly menu driven screen smoothly switches back and forth between the Editor, Compiler, Assembler and Linker.

The CoCo-C Compiler package includes BOTH CoCo 2 and CoCo 3 versions of ALL the programs listed above plus MORE! Compatible w/B&B RGBDOS

Never before has there been an offer like this for the Color Computer!

Requires 64K COCO 2 or 128K COCO 3

Send check or money order to:

Introductory Offer

**Infinitum Technology**

Only \$59.95

P.O. Box 356  
Saddle River, N.J. 07458

Plus \$4.00 shipping & handling

914-356-7688



Feature Program

# GRID WAR

## by MIKE SWITZER

**A**lthough *Grid War* plays pretty much like the board game *Battleship*, it does have one great advantage: it keeps the players honest. As with *Battleship*, each player hides four naval vessels on a grid and takes pot shots at the other player's ships. The first player to sink all of the other's ships wins. But with the board game, it is possible for clever players (a light term for cheaters) to move their ships, making it nearly impossible for the other player to win. Since *Grid War* is played on the Color Computer, it is impossible to move the ships during the battle.

*Grid War* is a two-player game designed for the CoCo 3; the game requires two joysticks. To play the game, first enter the listing and save the program to tape or disk. The game uses the high-speed poke (POKE 65497, 0), so be sure the computer is set to low speed (POKE 65496, 0) before you save it. With a copy of the game tucked safely away, enter RUN.

After the title screen appears, the two players are prompted to enter their names. The next screen displays the playing grid, along with the four ships and the number of grid spaces each requires. The current ship to be positioned is indicated in the middle at the bottom of the screen. While his opponent is not watching, each player uses the joystick to place his ships on the grid. To do this, move the block cursor to the desired location and press the fire button. You are asked whether the ship should be oriented vertically or horizontally. Press V or H accordingly. If the ship's placement creates no conflicts with other ships or the grid's borders, the ship is drawn and the next vessel to be placed appears.

Once all the ships are in place, you are asked if the positioning is correct. If you want to change where a ship is located, now is the time to do it.

After both players have placed their ships, another screen appears showing both players' grids side by side. Each player in turn uses his joystick to select a square on his opponent's grid. When he presses the fire button, the screen displays either a Hit or a Miss, and the other player gets a turn. Play continues until all the ships on one grid have been located and sunk. At that point, the ships on both grids are shown and the game is over.

*Mike Switzer is an electrical engineer who has worked with the CoCo and other computers for 10 years. His hobbies include drag racing, model railroading and watching the Buffalo Bills.*

**CoCo 3**

**The Listing:** GRIDWAR

- 1 \* GRID WAR
- 2 \* BY MIKE SWITZER AND
- 3 \* CECILIA FORLINTI
- 4 \* COPYRIGHT (C) 1992
- 5 \* BY FALSOFT, INC.
- 6 \* RAINBOW MAGAZINE

```

7 ON BRK GOTO 840:POKE65497,0
8 *
9 *SET VARIABLES
10 HSCREEN 2:PALETTE 2,15:PALETTE
E,4,56:PALETTE 7,52
15 DIM GC(2,10,10),GS(2,10,10)
20 HBUF 1,152:HBUF 2,152
25 S(1)="-BLSU4ER14FRFRFD2GLGL
GLGL14HU4":NS(1)="-CRUISER"
30 S(2)="-BLSUEUERER23FRFRFD2G
LGLGL23HLHUHU":NS(2)="-DESTROY
ER"
35 S(3)="-BLSUERERER32FRFRFD
26LGLGL32HLHLHUHU":NS(3)="-BAT
TLESHIP"
40 S(4)="-BLSU2ER12ER32DR10F06
GL56HU4":NS(4)="-CARRIER"
45 S(5)="-U3R10FRFRFGLGL10U3"
50 S(6)="-UEER17FRFRFGLGL17HUHU"
55 S(7)="-ERERER22FRFRFGLGL22H
LHLH"
60 S(8)="-ER9ER22DR5FD3L40U3"
98 *
99 *DRAW TITLE PAGE
100 DL=106:GOSUB1000
105 HGET(160,60)-(172,72),2:HDRA
W"BM166,66NEINFING1H1":HGET(160,
60)-(172,72),1
110 H=5:V=3:GOSUB1100:FOR TD=1 T
O 300:NEXT H=1:V=4:GOSUB1200:H=0
:GOSUB1100:FOR TD=1 TO 500:NEXT:
FOR H=2 TO 4:GOSUB 1200:NEXT H
115 HCOLOR,1,2:HDRAW"S2BM47,18R2
2F8D8L8U4H4L14G4D32F4R14E4U4L8U8
R16D16G8L22H8U40E8"
120 HDRAW"BM45,52R30F8D16G8L14F2
4L8H24D24L8U56F8R18F4D8G4L8U16"
125 HDRAW"BM49,86R24D8L8U40R8D8L
24U8R8U40L8U8"
130 HDRAW"BM45,120R30F8D40G8L30U
56F8R18F4D32G4L18U40"
135 HDRAW"BM256,32R8D36E12F12U36
R8D56L8U12H12G12D12L8U56"
140 HDRAW"BM264,70R8F16D40L8U24L
24D24L8U40E16D8D8R4F12D4L24U4E12"
145 HDRAW"BM256,108R32F8D16G8L16
F24L8H24D24L8U56F8R20F4D8G4L20U
16":HDRAW"54"
150 HPRINT(8,22),"WRITTEN BY MK
E SWITZER":FOR T=1 TO 5000:NEXT
T:GOSUB 1400
198 *
199 *ENTER PLAYERS NAMES
200 HCOLOR,2:FOR P = 1 TO 2:HPR
INT(2,22),"PLAYER #":HPRINT(10,2
2),STR$(P):HPRINT(12,22),"S NAM
E":SOUND 100,5
205 HT=20:VT=22:GOSUB 1300:PS(P)
=-TS:GOSUB 1400:NEXT P:P=1
298 *
299 *ENTER COORDINATES
300 GOSUB1000:H=28:V=51:Y=4:GOSU
B 320:HPRINT(6,6),"A":HPRINT(3,8
),NS(4):HPRINT(2,9),"(5 GRIDS)"
305 H=24B:Y=3:GOSUB 320:HPRINT(3
,6),"B":HPRINT(29,8),NS(3):HPR
INT(29,9),"(4 GRIDS)"
310 H=46:V=107:Y=1:GOSUB 320:HPR
INT(6,13),"C":HPRINT(3,15),NS(1
):HPRINT(2,16),"(2 GRIDS)"
315 H=256:Y=2:GOSUB 320:HPRINT(3
,13),"D":HPRINT(29,15),NS(2):H
PRINT(29,16),"(3 GRIDS)":GOTO 32
5
320 HDRAW"BM"+STR$(H)+"",+STR$(V
)+S(Y):HPAINT(H+2,V),4,8:RETURN
325 L=LEN(P):HPRINT(20-(L/2)
,1),SPR1:Y=1
330 HPRINT(2,22),"COORDINATES FO
R":HPRINT(18,22),NS(Y):SOUND100
,5
335 IF P=1:THENJ=0 ELSE J=2
340 GOSUB1500:GOSUB1400
345 HPRINT(2,22),"DIRECTION <H O
R V>?":SOUND100,5
350 HT=22:VT=22:GOSUB1600:IFLS="
H"THENDS(P,Y)=-"B" ELSE DS(P,Y)=""
1"
355 HS(P,Y)=H:VSP(P,Y)=VS(P,Y)
398 *
399 *CHECK IF SHIPS FIT
400 IF DS(P,Y)="" THEN 425
405 IF H+Y>9 THEN 445
410 FOR CF=H TO H+Y:IF GC(P,CF,V)
<>0 THEN 445
415 NEXT CF
420 GOTO 455
425 IF V+Y>9 THEN 445
430 FOR CF=V TO V+Y:IF GC(P,H,CF)
<>0 THEN 445
435 NEXT CF
440 GOTO 455
445 GOSUB 1900
450 GOTO 330
455 HDRAW"A"+DS(P,Y)+"BM"+STR$(
H+2)+106+"",+STR$(V+12)+30)+S
(Y)
460 GOSUB 1700
465 GOSUB1400:IF E=1 THEN 470 EL
SE Y=Y+1:IF Y<5 THEN 330
470 E=0:HPRINT(2,22),"IS THIS PA
TERN OK <Y/N>?":SOUND 100,5
475 HT=28:GOSUB1600:IFLS="N" THE
N 500
480 IF P<>2 THEN P=2:HDRAW"A0":G
OTO300
485 GOSUB1800:P=1:GOTO600
490 *
499 *REPLACE SHIP
500 GOSUB1400:HPRINT(0,22),"CHAN
GE WHICH SHIP <A>,<B>,<C>, OR <D
>?"
505 HT=39:GOSUB 1600
510 IF LS="C" THEN Y=1 ELSE IF L
S="D" THEN Y=2 ELSE IF LS="B" TH
EN Y=3 ELSE Y=4
515 HCOLOR,2:DS(P,Y)=DS(P,Y):H
=HS(P,Y):V=VS(P,Y):SV=0
520 HDRAW"A"+DS(P,Y)+"BM"+STR$(
H+2)+106+"",+STR$(V+12)+30)+S
(Y)
525 HCOLOR,2:GOSUB 1700:GOSUB 1
400:E=1:GOTO330
598 *
599 *FIRE SHOT AT GRID
600 GOSUB1400:SOUND 100,5:HPRINT
(2,22),PS(P):L=LEN(PS(P))+1:HPRI
NT((L+2),22),"SELECT GRID"
605 IF P=1 THEN G=2:DL=176:J=0 E
LSE G=1:DL=136:J=2
610 GOSUB 1500
615 IF GS(P,H,V)="" OR GS(P,H
,V)="" THEN GOSUB 1900:GOTO 600
620 IF GC(G,H,V)=0 THEN GOSUB 11
00:GS(P,H,V)=""
625 GOSUB 1200:GS(P,H,V)=""
630 IF GC(G,H,V)=1 AND K(P,1)=1
635 IF GC(G,H,V)=1 AND K(P,1)=2
THEN 700
640 IF GC(G,H,V)=2 AND K(P,2)=3
THEN 700
645 IF GC(G,H,V)=3 AND K(P,3)=4
THEN 700
645 IF GC(G,H,V)=4 AND K(P,4)=5
THEN 700
650 GOTO 705
698 *
699 *DID SHOT SINK SHIP?
700 H(P)=H(P)+1:GOSUB2000
705 FOR TD=1 TO 1000: NEXT TD:IF
H(P)=4 THEN 715 ELSE IF P=1 THE
N P=2 ELSE P=1
710 GOTO 600
715 GOSUB 1400:HCOLOR 8,2:HPRINT
(2,21),"YOU HAVE SUNK ALL OF":HP
RINT(23,21),PS(G):L=LEN(PS(G)):H
PRINT(23+L,21),"S SHIPS":FOR TD
=1 TO 1000:NEXT TD:GOTO 800
798 *
799 *END OF GAME
800 FOR P=1 TO 2
805 IF P=1 THEN DL=36 ELSE DL=17
6
810 FOR Y=1 TO 4
815 HCOLOR 1,2
820 HDRAW"A"+DS(P,Y)+"BM"+STR$(
HS(P,Y)*12+DL)+",+STR$(VS(P,
Y)*12+30)+S(Y)
825 NEXT Y,P
830 HPRINT(2,23),"THIS GAME IS O
VER. PRESS ANY KEY..."
835 YS=INKEY$:IF YS="" THEN 835
840 CLS:POKE 65496,0:END
990 *
999 *DRAW GRID
1000 HCLS2:HCOLOR,2
1005 FOR V=DL-6 TO DL+114 STEP 1
2:HLIN(6,V,24)-(V,144),PSET:NEXT
1010 FOR H=24 TO 144 STEP 12:HLI
NE(DL-6,H)-(DL+114,H),PSET:NEXT:
RETURN
1098 *
1099 *MISS ROUTINE
1100 FOR R=3 TO 6 STEP 3:HCIRCLE
(H+12,DL,V+12+30),R,8:NEXT R:PM=
31:GOSUB1215
1105 RETURN
1198 *
1199 *HIT ROUTINE
1200 HH=H+12:DL:VH=V+12+30:HCIRC
LE(HH,VH),2,4:HPAINT(HH,VH),4,4:
PM=1:GOSUB1215
1205 HCIRCLE(HH,VH),2,1:HCIRCLE(
HH,VH),4,1:HPAINT(HH,VH-2),1,1:G
OSUB1215
1210 HCIRCLE(HH,VH),4,7:HCIRCLE(
HH,VH),5,7:HPAINT(HH,VH-4),7,7:P
M=31
1215 PLAY"01L255V31":FOR M=1 TO
PM:PLAY"10N3N5N4N2N6V":NEXT M:
RETURN
1298 *
1299 *ENTER TEXT ROUTINE
1300 TS=""
1305 TS=INKEY$:IF LS="" THEN 130
5
1310 IF LS=CHR$(8) THEN LS=RIGHT
$(TS,1):TS=LEFT$(TS,LEN(TS)-1):H
COLOR,2:HPRINT(HT,VT),LS:HCOLOR
8,2:HT=HT-1:GOTO1305
1315 IF LS=CHR$(13) THEN RETURN
1320 TS=TS+LS:HT=HT+1:HPRINT(HT,
VT),LS:GOTO1305
1398 *
1399 *ERASE TEXT
1400 HLIN(0,176)-(319,191),PRES
ET,BF:RETURN
1498 *
1499 *LOCATES THE CURSOR WITH TH
E JOYSTICK
1500 HD=V=30
1505 JS=JOYSTK(0):HH=JOYSTK(J)*1
.8:V=V-JOYSTK(J+1)*1.8
1510 HD=DL+(INT((HH/12)*2)):V=30+(
INT((V/12)*2))
1515 HGET(H-6,V-6)-(H+6,V+6),2
1520 HPUT(H-6,V-6)-(H+6,V+6),1,A
ND
1525 HPUT(H-6,V-6)-(H+6,V+6),2,P
SET
1530 IF BUTTON(J)<1 THEN 1505
1535 H=(H-DL)/12:V=(V-30)/12:RET
URN
1540 LS=INKEY$:IF LS="" THEN 154
0 ELSE HPRINT(HT,VT),LS:RETURN
1598 *
1599 *INPUT COORDINATES TEXT
1600 LS=INKEY$:IF LS="" THEN 160
0 ELSE HPRINT(HT,VT),LS:RETURN
1698 *
1699 *ASSIGN SHIP TO GRID NUMBER
1700 IF DS(P,Y)="" THEN 1715
1705 GC(P,H,V)=SV:GC(P,H+1,V)=SV
:IF Y=1 THEN RETURN ELSE GC(P,H)
+2:V=SV:IF Y=2 THEN RETURN
1710 GC(P,H+3,V)=SV:IF Y=3 THEN
RETURN ELSE GC(P,H+4,V)=SV:RETUR
N
1715 GC(P,H,V)=SV:GC(P,H,V+1)=SV
:IF Y=1 THEN RETURN ELSE GC(P,H,
V+2)=SV:IF Y=2 THEN RETURN
1720 GC(P,H,V+3)=SV:IF Y=3 THEN
RETURN ELSE GC(P,H,V+4)=SV:RETUR
N
1798 *
1799 *DRAW BOTH GRIDS
1800 DL=36:C=4:GOSUB1000:L=LEN(P
$(1)):HPRINT(11-(L/2),1),PS(1):F
OR X=1:TO:04:ON X GOSUB1805,1810,18
15,1820:NEXT X:GOTO 1825
1805 H=30:V=152:Y=5:GOSUB2100:RE
TURN
1810 H=48:V=152:Y=6:GOSUB2100:RE
TURN
1815 H=75:V=152:Y=7:GOSUB2100:RE
TURN
1820 H=110:V=152:Y=8:GOSUB2100:R
ETURN
1825 DL=176:GOSUB1005:L=LEN(PS(2
)):HPRINT(28-(L/2),1),PS(2):FORX
=1:TO:04:ON X GOSUB1830,1835,1840,1
845:NEXT X:RETURN
1840 H=170:V=152:Y=5:GOSUB2100:R
ETURN
1835 H=188:V=152:Y=6:GOSUB2100:R
ETURN
1840 H=215:V=152:Y=7:GOSUB2100:R
ETURN
1845 H=250:V=152:Y=8:GOSUB2100:R
ETURN
1898 *
1899 *TRY AGAIN ROUTINE
1900 GOSUB 1400:HPRINT(2,22),"TR
Y AGAIN":SOUND 200,5:SOUND 100,5
:FOR TD=1 TO 1000:NEXT TD:GOSUB
1400:RETURN
1998 *
1999 *SUNK SHIP ROUTINE
2000 FOR X=1 TO GC(B,H,V):PM=31:
GOSUB 1215:NEXT X
2005 GOSUB 1400:HPRINT(10,22),"Y
OU SUNK THE":HPRINT(23,22),NS(G
C(H,H,V)):C=2:ON G GOSUB 2010,20
15:RETURN
2010 ON GC(G,H,V) GOSUB 1805,181
0,1815,1820:RETURN
2015 ON GC(G,H,V) GOSUB 1830,183
5,1840,1845:RETURN
2098 *
2099 *DRAW SHIP AT COORDINATES
2100 HDRAW"BM"+STR$(H)+"",+STR
$(V)+S(Y):HPAINT(H+2,V),C,8:RET
URN

```

SNOWFLAKE FROM COVER



Flakes

```
icons/icon.Flake
94
7
80
24
2
3
```

Figure 1: AIF:Flake Contents

```
iniz w1
wcreate /w1 -s=07 00 00 80 24 03 02 02
merge /dd/sys/stdfonts>/w1
shell i=/w1&
```

Figure 2: Commands for a Type 7 Window

save\* >Flake\_Maker

shown in Listing 2 (Flakes) is a replacement for the Flake\_Maker procedure. Flakes selects a black background and makes the Flake Maker system suitable for use as a screen saver. It is designed to be used in conjunction with the Flake procedure. The program in Listing 3 creates an icon file for using Flake Maker with Multi-Vue; the AIF for use in this fashion is shown in Figure 1.

Up and Running

To get started, first create a Type 7 graphics window by entering the lines shown in Figure 2. Alternatively, enter the command lines from Figure 2 in a standard OS-9 procedure file, then execute that file. When executed, these commands define Device /w1 as a 4-color graphics window having a resolution of 640 by 192 pixels. The last line starts an OS-9 shell on that window. To move to the window, simply press the CLEAR key.

With the proper window running, start BASIC9, requesting an 8K workspace (enter basic9 #8k). Now tell BASIC9 you want to edit the procedure Flake\_Maker by typing e Flake\_Maker and pressing ENTER.

In the listing for Flake\_Maker, you'll notice each line begins with a four-digit hexadecimal number and several spaces. These numbers are used by BASIC9 for debugging, and they show the number of bytes between the beginning of the procedure and the beginning of each line. Do not enter these numbers or the leading spaces when entering a BASIC9 procedure.

To enter each line, first press the space bar (which tells the editor you are inserting a line), then type the text of the line itself, pressing ENTER when you reach the end. When you have entered the last line of Flake\_Maker, enter the letter q on a line by itself to quit the editor. This returns you to BASIC9's B: prompt. Enter Flake in the same manner, checking each line for errors as you go along.

If BASIC9 reports any errors after you exit the Edit mode by entering q, you'll have to correct the errors by deleting and retyping the erroneous line. BASIC9 attempts to tell you where to find each error by reporting the memory offset to the line containing the error along with the error number. For example, suppose you see the following when you exit the editor:

```
00BC ERROR #069
```

The first number is the Hex memory offset from the beginning of the procedure to the line containing the error. The last number is the actual error number. Restart the Edit mode by entering e followed by a space and the name of the procedure you want to correct. When in the Edit mode, you can enter 1 to list the first line of the procedure. Press ENTER to scroll through the procedure one line at a time. When the numbers at the left of the listed line are greater than the number shown with the error, you have found the line after the one with the error. Move back one line by typing a minus sign (-) and pressing ENTER. Enter d to delete the line, then press the space bar once and retype the line correctly.

Before you run Flake\_Maker, save both procedures to disk by entering:

This command line saves both procedures in a file called Flake\_Maker in the current data directory. Then enter run flake\_maker to start the program. Just answer the question about background color, and Flake Maker will go to work drawing random snowflakes on the screen.

Program Options

Flake Maker is pretty nifty by itself. However, you may want to use it as a screen saver or from within Multi-Vue. Let's look at how you can do this.

To build Flake Maker as a screen saver, first delete all the procedures from the BASIC9 workspace by entering kill\* (but make sure you first save them to disk as described above!). Now enter Flakes from Listing 2. As I said before, Flakes is designed to work with Flake and replaces the Flake\_Maker portion of Listing 1. To get Flake into the BASIC9 workspace, enter load flake\_maker to load the saved set of procedures from disk, then kill the single Flake\_Maker procedure by entering kill flake\_maker. This command kills only the first procedure, leaving Flake in memory. Save the new procedures together as before, entering save\* >Flakes.

To test Flake Maker as a screen saver, enter run flakes. Press any key to stop the program.

Now you'll probably want to use your new screen saver from the OS-9 command line rather than having to run BASIC9 every time. To do this, you must pack the procedures it uses by entering pack\*. This tells BASIC9 to compile the screen saver and save the executable program in your current execution directory. (Never pack a procedure before you have saved it to disk. Once a procedure is packed, you cannot edit it — you'll have to revert to the unpacked version to make modifications.)

Now your CoCo can draw snow flakes on any graphics window (though Flake looks best in a Type 7 window) simply by entering flakes at the OS-9 prompt. It is important to note, however, that runb, inkey and gfx2 must be in memory or in your current execution directory. It is a good idea to merge all of these, along with syscall1, into one file to conserve system memory. To do this, first change the current data directory to the directory where runb is located (usually /dd/CMDS) by entering chd /dd/cmds. Then save a copy of the original runb file by renaming it as runb.org:

```
rename runb runb.org
```

Now, put runb, gfx2, inkey and syscall1 into one file by entering

```
merge runb.org gfx2 inkey sysca1
1 >runb
```

Finally, make the new merged runb executable by entering attr runb e pe.

If you follow these steps, you'll have a copy of the original runb file for later use, as well as a merged copy for running packed BASIC9 programs. If you have limited disk space, you can delete the original files and keep only the new runb file (as long as you are working with a copy of the system disks and not the originals from Tandy).

If you use Multi-Vue, it is easy to set up

Flake Maker so that it executes when you click on an icon. Enter Listing 3, then save it and run it. This procedure creates an icon file in the /dd/CMDS/ICONS directory. Once that icon is created, exit BASIC9 by entering bye at the B: prompt.

Use your favorite text editor or the OS-9 build command to enter the application-information file (AIF) shown in Figure 1, then save this file to disk in the data directory of your choice. A quirk in Multi-Vue is that the system does not recognize a new AIF unless the disk is scanned from its root directory. You can force a scan by clicking on the drive icon at the left side of the GShell window, then clicking on each folder icon until you get to your new AIF. You should see a snowflake icon in the directory. If you see the document icon, there is something wrong with the icon file, the AIF or the program file. Check the attributes of the program file and the icon file. The public- and owner-execution attributes must be set (attr should return -e -r ewr). List the application-information file and check to make sure the lines exactly match those in Figure 1. Don't forget to enter the blank lines since these tell GShell to use default

information when starting the program.

On a related note, one of the things I like most about Multi-Vue is that it does a good job of setting up the appropriate window types for the programs I use. I've heard many users have had trouble getting Multi-Vue to work properly. I would be more than happy to help anyone who writes to me asking for help with Multi-Vue. If I get enough response, perhaps I'll work on an article that addresses the most common problems.

In the meantime, enjoy the snowflakes. Though they won't melt on your tongue, they are definitely standard fare for the season.

-----  
Bill Budenholzer has been a CoCo user since 1981 and has learned to program in both BASIC and C. His experience with OS-9 has helped him gain the skills necessary to support Unix workstation users. He can be contacted at 7115 Lanham Ave., St. Louis, MO 63143-2511. Please include an SASE when requesting a reply.

OS-9 Level II

Listing 1: Flake\_Maker.b09

```
PROCEDURE Flake_maker
0000 DIM back:STRING
0007 DIM d1:INTEGER
000E DIM key:STRING
0015 key=""
001C RUN gfx2("clear")
0029 RUN gfx2("curxy",.15,.10)
003C PRINT "Do you want a (B)lack or (W)hite background?"
006C RUN gfx2("curxy",.15,.11)
007F INPUT back
0084 REPEAT
0086 RUN flake(back)
0090 FOR d1:=1 TO 10000
00A1 NEXT d1
00AC RUN inkey(key)
00B6 UNTIL key<"*"
00C1 RUN gfx2("color",.1,.0)
00D4 END

PROCEDURE Flake
0000 (* Copyright 1992 by William G. Budenholzer. All Rights Reserved *)
0043 (* Draws random snow flakes on graphics screen *)
0086 (* back sets the background color *)
00A8 PARAM back:STRING
00B2 (* hx and hy are the coords for the hexagons *)
00E1 DIM hx,hy:REAL
00EC (* hex_sz is the size of the hexagon *)
0113 DIM hex_sz:REAL
011A (* hexcnt counts the hexagons *)
013A DIM hexcnt:INTEGER
0141 (* splen is the length of a spike *)
0165 DIM splen(10):REAL
0171 (* spcnt counts the number of spikes *)
0190 DIM spcnt:INTEGER
019F (* num_spikes is the total number of spikes on one side of arm *)
01E7 DIM num_spikes:INTEGER
01E7 (* num_flakes is the number of flakes *)
020F DIM num_flakes:INTEGER
0216 (* col is the random color of the flakes *)
0241 DIM col:INTEGER
0248 (* cnt counts the number of spikes *)
0260 DIM cnt:INTEGER
0274 (* x1 and y1 are the end points of the current arm *)
02A9 DIM x1,y1:REAL
02B4 (* cx and cy are the center of the whole flake *)
02E5 DIM cx,cy:REAL
02F0 (* angle is the angle of the flake's arm *)
031B DIM angle:REAL
0322 (* rot_angle is the rotation angle of the whole flake *)
035A DIM rot_angle:REAL
0361 (* xscale and yscale correct the length of the arms to make a round
flake *)
03AD DIM xscale,yscale:INTEGER
03B8 (* radius is the total radius of the flake *)
03E5 DIM radius:REAL
03EC (* sx and sy are end points for the spikes *)
0419 DIM sx1(9),sy1(9),sx2(9),sy2(9):REAL
0440 DIM sx3(9),sy3(9):REAL
0455
0456 (* the circle plot routines won't work in radians *)
048A DEG
048C
0480 (* if back = "B" if we want a black background *)
04BE IF back="B" OR back="b" THEN
04D3 RUN gfx2("palette",.0,$0B)
04FE RUN gfx2("palette",.1,$3F)
04FF (* otherwise we get a white background *)
0528 ELSE
052C RUN gfx2("palette",.0,$3F)
0542 RUN gfx2("palette",.1,$09)
0558 ENDF
055A RUN gfx2("palette",.2,$1A)
0570 RUN gfx2("palette",.3,$1B)
0586 RUN gfx2("color",.1,.0)
0599 col:=1
05A0 RUN gfx2("clear")
05A0 RUN gfx2("border",.0)
05BE RUN gfx2("curroff")
05CC xscale:=2
05D3 yscale:=1
05DA (* Reseed the random number generator *)
0602 cnt:=RND(-(VAL(MID$(DATE$,16,2))))
0611
0612 (* make 2 to 5 flakes *)
062A FOR num_flakes:=1 TO 2+RND(3)
0641 radius:=RND(20)+20
```

```

064E col:=col+1
0659 IF col>3 THEN
0665 col:=1
066C ENDF
066E RUN gfx2("color",col)
0680 rot_angle:=RND(30)
0689 (* find center of next flake *)
06A8 (* don't let flake go off screen *)
06CB cx:=RND(540)
06D5 WHILE cx<100 DO
06E2 cx:=RND(540)
06EC ENDF
06F0 cy:=RND(130)
06F9 WHILE cy<60 DO
0706 cy:=RND(130)
070F ENDF
0713
0714 (* build an arm *)
0726 (* make 5 to 9 spikes *)
073E num_spikes:=RND(5)+4
074C (* dist1 sets the spacing for the spikes *)
0777 dist1:=radius/(num_spikes+1)
078F dist2:=0
0790 FOR spcnt:=1 TO num_spikes
07A1 splen(spcnt):=RND(radius/2)
07B2 NEXT spcnt
07BD
07BE FOR angle:=0 TO 300 STEP 60
07D7 FOR spcnt:=1 TO num_spikes
07E8 (* dist2 is the accumulated distance from the center *)
081F dist2:=dist2+dist1
082B sx1(spcnt):=cx+COS(angle+rot_angle)*(dist2*xscale)
0840 sy1(spcnt):=cy-SIN(angle+rot_angle)*(dist2*yyscale)
085F sx2(spcnt):=sx1(spcnt)+COS(angle+rot_angle+60)*(splen
(spcnt)*xscale)
0897 sy2(spcnt):=sy1(spcnt)-SIN(angle+rot_angle+60)*(splen
(spcnt)*yyscale)
08BF sx3(spcnt):=sx1(spcnt)+COS(angle+rot_angle+300)*(splen
(spcnt)*xscale)
08E8 sy3(spcnt):=sy1(spcnt)-SIN(angle+rot_angle+300)*(splen
(spcnt)*yyscale)
0911 NEXT spcnt
0912 FOR cnt:=1 TO num_spikes
091D RUN gfx2("line",FIX(sx1(cnt)),FIX(sy1(cnt)),FIX(sx2(cnt)
)),FIX(sy2(cnt)))
095A RUN gfx2("line",FIX(sx1(cnt)),FIX(sy1(cnt)),FIX(sx3(cnt)
)),FIX(sy3(cnt)))
0986 NEXT cnt
0991
0992 (* draw main arms *)
09A6 x1=cx+COS(angle+rot_angle)*(radius*xscale)
09C0 y1=cy-SIN(angle+rot_angle)*(radius*yyscale)
09DA RUN gfx2("line",FIX(cx),FIX(cy),FIX(x1),FIX(y1))
09FA dist2:=0
0A02 NEXT angle
0A0D RUN gfx2("setdptr",FIX(cx),FIX(cy))
0A26
0A27 (* draw hexagons around middle of flake *)
0A51 FOR hexcnt:=1 TO 2+RND(3)
0A58 hex_sz:=RND(radius/3)
0A75 FOR angle:=0 TO 360 STEP 60
0A8E hx:=cx+COS(angle+rot_angle)*(hex_sz*xscale)

```

```

0AAB hy:=cy-SIN(angle+rot_angle)*(hex_sz*yyscale)
0AC2 IF angle<30 THEN
0ACF RUN gfx2("setdptr",FIX(hx),FIX(hy))
0AEB ELSE
0AEC RUN gfx2("line",FIX(hx),FIX(hy))
0B02 ENDF
0B04 NEXT angle
0B0F NEXT hexcnt
0B1A
0B1B NEXT num_flakes
0B26
0B27 RUN gfx2("curon")
0B34 END

```

Listing 2: Flakes.b09

```

PROCEDURE Flakes
0000 DIM back:STRING
0007 DIM d1:INTEGER
000E DIM key:STRING
0015 key:=""
001C RUN gfx2("clear")
0029 back:="B"
0031 REPEAT
0033 RUN flake(back)
003D FOR d1:=1 TO 10000
004E NEXT d1
0059 RUN inkey(key)
0063 UNTIL key<>" "
006E RUN gfx2("color",1,0)
0081 END

```

Listing 3: MakeIcon.b09

```

PROCEDURE MakeIcon
0000 (* Generates a Multi-View Icon File
0022 (* For Flakes
002F DIM path,byt:BYTE
003A DIM count:INTEGER
0041 PRINT "Creating Icon,Flake . . ."
005F CREATE #path,"dd/cmds/icons/icon.Flake":WRITE
0083 FOR count:=1 TO 144
0093 READ byt
0098 PUT #path,byt
00A2 NEXT count
00AD CLOSE #path
00B3 PRINT
00B5 SHELL "att /dd/cmds/icons/icon.Flake e pe"
00DC END
00DE DATA $00,$00,$00,$00,$00,$00,$00,$00,$01,$00,$00,$00,$00,$00,$01,$00
0122 DATA $00,$00,$00,$00,$04,$04,$00,$00,$01,$04,$11,$10,$41,$00,$00,$00,$45
0166 DATA $05,$41,$44,$00,$00,$14,$51,$14,$50,$00,$01,$54,$11,$10,$55,$00
01AA DATA $00,$01,$15,$51,$00,$00,$00,$00,$51,$14,$00,$00,$00,$15,$51,$15
01EE DATA $50,$00,$00,$00,$05,$40,$00,$00,$00,$05,$40,$00,$00,$00,$00,$15
0232 DATA $51,$15,$50,$00,$00,$00,$51,$14,$00,$00,$00,$01,$15,$51,$00,$00
0276 DATA $01,$54,$11,$10,$55,$00,$00,$14,$51,$14,$50,$00,$00,$45,$11,$01
02BA DATA $40,$00,$01,$04,$05,$40,$41,$00,$00,$00,$11,$10,$00,$00,$00,$00
02FE DATA $41,$04,$00,$00,$00,$00,$01,$00,$00,$00,$00,$00,$00,$00,$00,$00

```

DS-69B FROM COVER

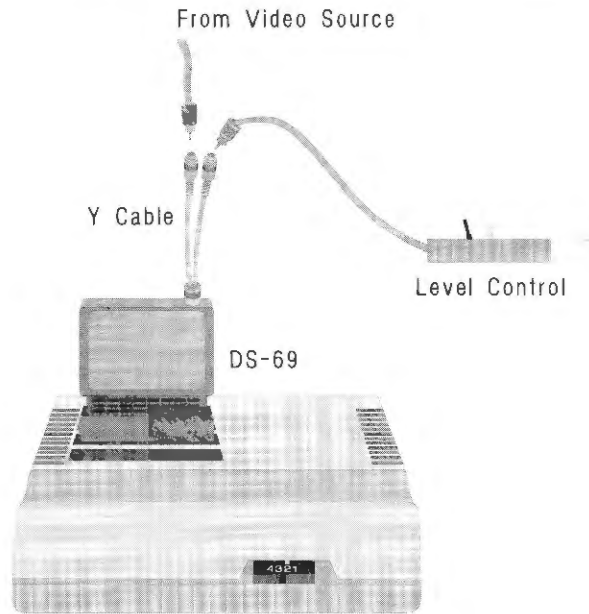


Figure 3

Once you have gathered the necessary parts, you should decide where you want the visible components — the switch, dimmer and cable — mounted on the project box. Drill holes of the proper size for these parts. Next, strip about two inches from one end of the shielded cable and about half an inch from the other end, then separate the negative shielding and insulated positive lead. Feed the shielded cable into the cable hole in the box, starting with the end from which you stripped two inches

of insulation. Now strip a very short length of insulation from the inner (positive) lead of the cable and solder this lead to one pole of the SPST switch. Strip a short length of insulation from each end of the standard wire and solder one end to the other pole of the switch.

Bend down the leads of the 1K fixed resistor, then slip one lead through Tab 1 on the potentiometer and the other lead through Tab 2. (Looking from the back of the potentiometer, with the tabs pointing up and away from you, Tab 1 is on the left and Tab 2 is in the center.) Bend the leads of the 1K resistor slightly so it does not fall out of the holes in the tabs. Solder the negative lead (the shielding) on the shielded cable to the center tab of the 5K potentiometer, along with the one end of the 1K resistor. Then solder the standard wire coming from the switch to Tab #1 of the potentiometer, along with the other end of the fixed resistor. After mounting the potentiometer and switch in the project box, all that remains is to solder the male RCA phono plug to the other end of the shielded cable.

As a construction note, the level control should work fine without the 1K fixed resistor. However, I added this resistor to

reduce the potentiometer's range to about 0 through 850 ohms, which makes it easier to manipulate the level at the brighter end.

Connecting the DS-69

To connect the level control to the DS-69B, first plug the male end of the Y cable into the digitizer. Then plug the video source into one of the female jacks on the Y cable and plug the level control into the other female jack.

The switch allows you to control whether or not the video signal being sent to the digitizer is affected by the level control. When the switch is open, the signal is unaffected. When you close the switch, however, you gain fine control over the brightness of the video signal.

Steve Ricketts is a mainframe operator who recently discovered the beauty of OS-9. He is interested in finding a way to use the DS-69B video digitizer under OS-9. Steve can be contacted at 39230 Hood St., Sandy, OR 97055. Please include an SASE when requesting a reply. He can also be reached on Delphi via the username STEVEFDX.

Part	Description	Radio Shack #
R1	1000 ohms	271-023
R2	5000 ohm pot	271-1714
SW1	submini SPST switch	275-645
PL1	shielded male RCA plug	274-339
Y1	shielded Y-adaptor	42-2436

In addition, you'll need some shielded cable (about 8 to 12 inches) and a short piece of standard insulated wire (22- or 24-gauge is fine).

Figure 2: Level Control Parts List

**Product Review**

**TasCOM: OSK Terminal Software**

Vaughn Cato's shareware terminal program, *OSTerm*, has long been a favorite among users of OS-9 Level II on the Color Computer 3. By taking advantage of the similarities between the windowing system for Level II and the *KWindows* system provided with the MM/1, Mr. Cato has made this popular program available for OS-9/68000. Although the name has changed to *TasCOM*, the program is enough like its predecessor that fans of *OSTerm* should be able to enjoy this program immediately. *TasCOM* does have its drawbacks, but its list of features is sufficiently impressive to outweigh the minor problems I encountered in the interface and documentation.

*TasCOM* has several features that make it convenient and fairly easy to use. The Autodialer function allows you to configure the program for each system you contact, and you can rapidly create a list of systems to be dialed. This is ideal for people who frequent a number of BBSs — *TasCOM* can be set to dial each system in turn, in round-robin fashion, until it finds one it can connect to.

The File-transfer section includes a convenient file picker that lets you scroll through directories with the arrow keys, selecting files to be transferred. You can use this to quickly create a list of files to be uploaded via Ymodem Batch, or to select a directory or specific filename for downloaded files. *TasCOM* also includes a fairly good ANSI terminal emulation that produces brightly colored menus when used

with a local bulletin board. As a final touch, *TasCOM*'s Remote mode allows others to dial your system and upload and download files. This easy-to-use feature provides a simple alternative to setting up the TSMon and LogIn utilities to handle remote logins.

I tested *TasCOM*, V1.0.1, on a 3-megabyte MM/1 running *KWindows* Edition 38. The installation process was quite simple; an install script on the distribution disk creates a /DD/TASCOM directory, copies the executable file and some shell scripts to /DD/CMDS, then copies a customizable setpalette script to /DD/TASCOM. With only five files, it is quite simple to install the system by hand if your configuration is different.

The shell scripts are a nice touch since they provide easy customization. The *tas* shell script used to start *TasCOM* sets the default directory to /DD/TASCOM, runs the setpalette script, creates a new window, then runs *TasCOM* in that window. This approach makes it relatively easy to customize the directory, the screen colors, the modem port and other parameters simply by editing the appropriate script.

I was a little disappointed that *TasCOM* did not by default use /dd/COM/TASCOM for its custom files, following a convention used by many OS-9 Level II terminal programs. But I found it easy to change this. The only problem I had installing *TasCOM* was that the files on the disk are readable only by the superuser, so I was forced to login to my system as root in order to read the files. Similarly, the install script did not set Public Read and Public Execute attributes for the files it installed, so I was forced to do this manually.

You can use ALT keys to control the program, allowing you to send any charac-

ter to the host. However, it did not allow me to send a NULL character. Overall, *TasCOM* is easy to use, although the interface is somewhat quirky. I believe this comes from a general inattention to minor details. For example, pressing ALT-/ brings up a list of the available commands, but you cannot select a command directly from that list. You must first press a key to get rid of the list. Similarly, the Autodialer provides no simple short cut for selecting and dialing a single system; you have to create a list with only one system on it, then dial. It took me several minutes of searching to discover how to edit the Autologin macros.

I also found some omissions and problems. I could find no way to get *TasCOM* to do end-of-line conversions when downloading or uploading ASCII files. This is a common feature of many terminal programs and is practically a necessity when transferring files between dissimilar systems. Also, the fact that *TasCOM* always has its default directory set to /DD/TASCOM initially was bothersome when transferring files — you have to manually change the directory every time you start the program. *TasCOM*'s VT100 emulation did not function correctly when used with full-screen programs on a Unix machine. However, the ANSI emulation worked acceptably.

A number of unexplained things happened while I was using the program, though it is difficult to tell whether they were faults with *TasCOM*, the still sometimes errant system software, or other programs that were running on the system simultaneously. Whenever a file transfer terminated, for example, garbage appeared on another window. This was sufficiently reproducible that it seems certain to be a bug in *TasCOM*.

There isn't much to say about the documentation, largely because there *isn't* much documentation. The 10-page manual gives little more than a list of the options available in each part of the program. There are no descriptions of the terminal emulations, particularly in the critical area of which keyboard keys correspond to some of the extended VT100 keys. There is no table of contents, and the crunched layout seems to emphasize saving paper over ease of reading. The fact that the manual instructs the user to press keys on the CoCo 3 but not on the MM/1 keyboard did not help to make a good first impression.

Overall, I find *TasCOM* sufficiently easy to use. I will probably continue to use it on an occasional basis, though I will continue to look for a program that does a better job of VT100 emulation, provides better support for file transfers, and sports a stronger interface. People who frequently dial a number of bulletin boards will appreciate *TasCOM*'s autodialing capabilities and its functional ANSI emulation. People who transfer files to and from friends' computers will appreciate the Remote mode. If the author takes the time to polish the user interface and address the numerous minor problems, *TasCOM* could be an excellent terminal program useful to a broad audience. As is, *TasCOM* is a functional program that is comfortable once you understand how to use it; and it has enough features to be a good "second" terminal program for a variety of users. (CoCoPRO!, 1334 Byron Ave., Ypsilanti, MI 48198, 313-481-3283; \$39.95.)

— Tim Kientzle

# star Plug 'n' Go for Your CoCo!



**7 Color Printer!**

## star NX-1020

This CoCo compatible NX-1020 system sets new standards in color printer performance... 225 cps, 4 NLQ fonts including Script, plus a high speed draft font; but the enhancements don't stop there. Add a 16k buffer, a special quiet mode, top feed, bottom and rear tractor, and the list goes on. Seven on-demand colors, 8 color graphic modes, Epson and IBM emulation for maximum software compatibility. Virtually everything desired in a printer is here — speed, color and versatility at an affordable price with a 2 year warranty.

**Our Plug'n'Go for the Coco system includes:**

- NX-1020 Multi Font Color Printer
- Blue Streak Ultima
- Software Support Disk
- Color Graphics Utilities

**\$239<sup>95</sup>**  
+ \$10 Shipping & Insurance



## star NX-1001

This CoCo compatible NX-1001 system is fully featured with 4 NLQ plus a draft font, 10 character sizes from subscript to quadruple size, 4k buffer, 180 cps, friction and tractor feed, and much much more. Backed by a 2 Year warranty. Epson and IBM emulation modes for maximum software compatibility. A performer so versatile you may never exhaust it's creative possibilities!

**Our Plug'n'Go for the Coco system includes:**

- NX-1001 Multi Font Printer
- Blue Streak Ultima
- Software Support Disk

**\$188<sup>88</sup>**  
+ \$10 Shipping & Insurance



## The Ultimate Serial to Parallel Interface!

### The Blue Streak Ultima

**7 Switchable Baud Rates - 300 thru 19200!**

- An interface cable that converts the serial output of a CoCo 1, 2 or 3 to a standard parallel format, compatible with modern parallel printers.
- Connecting the Ultima is as easy as plugging in the cable! The four pin din plugs into the serial I/O port of your CoCo and the other end, a 36 pin connector, connects to the parallel port of the printer.
- The Ultima is powered with the +5V supplied by most printers on pin 18. If your printer does not have +5V on pin 18 you'll need to add the power option when ordering.

**\$39<sup>95</sup>**  
+ \$2 Shipping  
POWERED VERSION  
ADD \$6.00

**STAR 24 WIRE PRINTER SYSTEMS ALSO AVAILABLE!**  
CoCo compatible • Color and Monochrome • Call for pricing

## Order Your System Today... Call (513) 885-5999

**DAYTON ASSOCIATES** of W.R. Hall, INC.  
9644 Quailwood Trail • Spring Valley, Ohio 45370



Visa & Mastercard Accepted.  
Ohio residents add 6.5% sales tax COD add \$4.00  
Shipping charges to Canada, P.R., HI, AK, APG, FPO are double. Triple charge to all other countries.  
Price and Specifications are subject to change without notice.

All Dayton Associate's products have a 30 day moneyback guarantee.

Epson is a registered trade mark of Seiko Epson Corp. IBM is a registered trade mark of International Business Machines Corp.



**MARTY GOODMAN**

**Dead CoCos and Odd Drives**

*When I play Arkanoid, the second player joystick won't work correctly. Using a different physical joystick does not fix the problem, and both joysticks work fine with other games and with BASIC programs. Also, my 512K CoCo just died. It comes up with a yellow screen with black vertical lines. What should I do? How can I change the disk drive ribbon connector on my JDOS drive to the disk drive connector on my Tandy drive?*

Owen I. Crabtree  
College Place, Washington

**512K On the Blink**

*I have a CoCo 3 with a dead 512K Disto memory board (it gives me a blank screen). When I took it to Radio Shack for repair, they stated the problem was with the third-party memory board and that they could not help me. Now I find that Disto/CRC is not available to help me, apparently having left the CoCo market. What can I do?*

Gil Brown  
San Luis Obispo, California

**A** First, remove the 512K board and re-install the four 4464-15 or equivalent memory chips that were originally in your CoCo 3. Does the computer work? If so, you have narrowed the problem down to the 512K board. Note that Radio Shack repair often won't service a computer with non-factory parts installed and may well have just returned it without much checking the moment they saw that Disto 512K board. To properly check a CoCo usually requires that you own a second one that is known to be good. You really should try testing that Disto board in a known-working CoCo 3.

If the 512K board is proven bad, you must determine whether the problem is a failed memory chip, or a problem with the board or its contacts. Suitable memory chips (256K-by-1, 120ns or 150ns) are cheap, costing a dollar apiece or less, so you might want to buy a few of them from a chip vendor and start substituting. Perhaps you have a bad chip. Other possible problems are shorts and/or cold solder joints on the 512K board. Carefully inspect the board, looking for bad solder joints. Try resoldering every point on the board (sounds tedious but actually takes only 5 or 10 minutes if you have some skill at fine soldering). You should consider swapping the chips out of a known-good 512K memory board and putting them on your suspect board to attempt to differentiate whether the problem is the memory chips or the board itself. Also, keep in mind that sometimes problems arise due to shoving the memory board too far into the connectors on the motherboard, or from the board shorting out to something beside or below it. Check for this.

If the 512K board tests good in another CoCo 3, try replacing the 68B09E chip (this requires that you desolder the 68B09E). Then try the GIME chip, though that is less likely to cause this type of problem. Good luck!

By the way, I will pay \$20 for a known-dead Color Computer, provided its GIME chip is intact and has not succumbed due to a lightning strike or some other obvious disaster. (I'm looking for spare parts to fix a dead computer of my own.) Also, if you have no way of testing the 512K upgrade board you have, I will test it for you if you send it to me with a postage-prepaid, self-addressed box. You can send it to me at 1633 Bayo Vista Ave., San Pablo, CA 94806. Finally, 128K CoCo 3's are often offered for around \$50 through the Classified Ads section in the CoCo SIG on Delphi.

**A** There's a bug in the code for Arkanoid. I don't know of any fix for it. Sorry.

Any number of things can cause a dead Color Computer, the most common being a bad 68B09E chip. Tied for second place are malfunctioning memory chips and a bad board. Other things to check include whether the 512K memory board is properly seated and whether the GIME chip is properly in its socket. Sometimes merely removing the GIME chip from its socket, spraying the socket with contact cleaner and re-inserting it cures the problem. For more information, see my response to Gil Brown in this issue.

All Color Computer disk-drive systems use the same 34-pin female edge connectors on their ribbon cables, with the same assignment of signals. All CoCo floppy controllers that plug into the CoCo or Multi-Pak Interface use the same assignment of pins on their edge connector that the 34-pin drive cable connects to. Thus, a disk drive system from J&M should plug in just fine to a Tandy disk controller, and vice versa. The only possible difference I can think of is that some edge connectors use a "key" — a sliver of plastic between the inside walls of the connector — to insure proper alignment of the connector. Replacement connectors without such keys are commonly available at electronics stores.

**Problems With The WIZ**

*I'm having trouble with some files I downloaded from the OS-9 SIG on Delphi that stated they were to go with The WIZ terminal program. Can you help me?*

Bruce Moore (THUNDERFINGERS)  
Sterling, Virginia

**A** The WIZ, a terminal program for OS-9, attempted (foolishly, I think) to introduce a new, incompatible variant of Xmodem. The altered protocol added an extra block (with information about the file) at the start of the file transfer. To download such files with other terminal programs, you'll need to strip off that first block. Such a "de-Wizzer" stripper program is available in the OS-9 SIG database. Alternatively, you can make use of the convenience the header offers by using The WIZ to download those files.

**Checking Things Out**

*Are there peeks to see if a CoCo 3 has 128K, 512K or 1MB of memory installed? Can I detect via software alone whether or not an RGB monitor is being used? Are there peeks to tell whether ADOS 3, Disk BASIC 1.0 or Disk BASIC 1.1 is in use in a given machine? Is there a way to use software to detect what kind of drive (35- or 40-track, single- or double-sided) is connected to the computer?*

Bob Williams (BAWILLIAMS)  
Ceder Hill, Missouri

**A** Memory size can be determined via software alone. The principle is as follows: In a 128K CoCo, if you tell the memory management unit (MMU) to se-

lect memory-block numbers \$00, \$10, \$20 or \$30, you will be selecting the same physical space in RAM because memory blocks ghost on top of each other in a 128K CoCo (see "New Clear-screen Routines," June 1992, Page 1). On the other hand, if you have 512K, selecting these blocks specifies four different physical areas of memory. How might you use this information to determine memory size? After power up, enter POKE &HFFA2,0. This maps Memory Block 0 into Slot 3 of BASIC's memory map at \$4000. Then enter POKE &H4000,173 to put a given value (173) into location \$4000. Now type POKE &HFFA2,&H10 to map Memory Block 16 (\$10) into the same part of BASIC's memory map. Finally, enter PRINT PEEK(&H4000). If the CoCo returns 173, you're using a 128K CoCo. If you get some other number, it means you have a 512K or 1-Meg CoCo.

You can use the same principle to distinguish between a 512K and 1-Meg CoCo 3 if you understand that, on a 512K machine, memory blocks \$40 through \$7F are the same as memory blocks \$00 through \$3F on a 512K CoCo 3. They are separate memory blocks on a 1-Meg CoCo 3.

When you have finished testing, it might be a good idea to return Location \$FFA2 to its default value of \$3A (just enter POKE &HFFA2,&H3A). It is also critical that you understand the general theory of how the MMU in the CoCo 3 works in order to really understand what is going on here. Since 1987, several articles covering this topic have appeared in THE RAINBOW, and the CoCo 3 service manual from Tandy also contains this information.

There is no way to check for the presence of an RGB or composite video monitor using software. (In my opinion, this is good, for such automatic checking would be fraught with problems had it been implemented.) Many authors get around this by asking the user which type of monitor he is using.

Checking for DOS types can be done by looking at the Disk BASIC ROM itself. Memory Locations \$C139 thru \$C191 contain the copyright message for Disk BASIC 1.1. Locations \$C126 thru \$C17D contain the copyright message for Disk BASIC 1.0. By peeking at what is there, you can compare the contents you find to what you would expect with either version of Disk BASIC. When it comes to variant disk ROMs, things can get a little trickier. ADOS comes in three main versions (ADOS, ADOS3 and Extended ADOS 3). Even worse, users can customize their own ROM sign-on message, and although they are advised to at least leave the version number in the same part of memory, they sometimes ignore this advice. You may want to consult with Art Flexser (author of ADOS) on Delphi about the best way to test for ADOS versions. Note that whatever test you use, you may need to have an exception-handling routine to deal with the case of unknown DOS versions where it finds one or another version of, say, JDOS, MYDOS or some other variant.

There's no simple way to test for what kind of a disk drive is hooked to the CoCo without putting a disk in the drive and trying to format, read from, and write to parts of that disk. With a drive merely connected to the CoCo but with no disk in it, there is no way the CoCo can tell what sort of drive it is. All the CoCo can do is tell a disk drive to read a track or sector, write a track or sector, move the head to position zero, or attempt to step the head one position up or down. While the CoCo does know (is told) when the head is at Track 0, it is not told when the head is at the last possible position. So if you step the head of

a 40-track drive from Track 0 to Track 79, it will repeatedly bang against the end stop. The CoCo has no way of knowing this is happening; for all it knows, the head is properly stepping in response to its command.

**The DMP-130 and the PC**

*I'm having trouble connecting a Tandy DMP-130 to an IBM computer via the parallel port. Any suggestions?*

Eric Stroh (COCOS12K)  
Thornton, Illinois

**A** This problem has come up many times with the DMP-130 series of printers. Tandy uses a different implementation of the Centronics parallel port than the international standard — Tandy put the printer-initialization line on Pin 33 of the printer's connector, while every one else uses Pin 31 and grounds Pin 33. Hence, when you use a normal cable to connect a DMP-130-series printer to an IBM PC or compatible, the printer totally locks up. There a number of ways to fix this. The easiest is to simply make a special modification in your parallel printer cable. Inside the cable connector, cut the wire that goes to Pin 33 of the printer connector. The printer should then work fine when used with this cable.

If you want to get fancy, you could reverse the wires that go to pins 33 and 31 on the cable. In that case, be very careful to label that cable "For use with DMP-130-series printers only." This latter approach preserves the printer-initialization option via the printer port — something not preserved with the simple wire cut. However, very few applications need to use the printer-initialization option, so it should not be a big deal if you disable it by doing the simple cable modification of cutting the single wire.

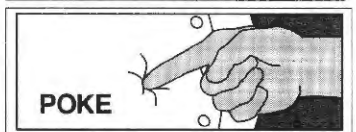
**Where is Track 17?**

*When you use software that supports 40- and 80-track drives, is the directory track still located on Track 17, or is it moved?*

Charles A. Marlow (CHARLESAM)  
Massapequa, New York

**A** All the software I've ever seen that supports 40- and 80-track drives keeps the directory on Track 17 to preserve compatibility. This is certainly true of ADOS and ADOS 3.

*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.*



Pressing the Reset button on the rear of the CoCo 3 does not clear the computer's memory. Yet turning the machine off and on again causes undue wear and tear. To completely reset the CoCo 3, erasing any programs in its memory, enter

```
POKE113,0:EXEC &#8BC1B
```

Feature Article

# OSK Part II

## A User's Experiences

by  
John  
Donaldson

Last month I shared some of my experiences in setting up OS-9/68000 and my impressions of how it works, explaining some differences between OSK and OS-9/6809 commands. This month we'll continue that path, looking deeper into the system. As you read this, however, bear in mind that the 68000-based computer I have is the MM/1. Certain specifics may differ if you are using a different OS-9/68000-based machine.

For those of you who are into C programming, the new Microware C compiler that comes with OS-9/68000 is good. Included on the disk with the C compiler are a sample program (makefile), make, the Microware assembler (r68), the linker (l68), and the debugger (debug). If you don't use the make utility under Level II on the CoCo 3, I recommend you do so with OSK. The sample makefile, shown in Figure 1, is straightforward and easily expanded.

The first thing you will notice is that it isn't like the makefile under C on the CoCo 3. To me, the OSK makefile is simpler and easier to use. No longer do I have to tell it which header files to use; it gets those from the C source code. Also, commands such as attr can be used from within the makefile. An example is making a C library. Just add

```
library.l: file.r file.r file.r
merge >file.r file.r file.r
```

right after the attr statement and add library.l in the link statement. Once you have made a library file, you could put it in the /dd/LIB directory and use it just like any other library file. I am putting together a library file that I call utx.l; in it are modules that emulate Unix calls. This will come in handy for porting C code from a Unix system to OSK.

Notice the line in the makefile that starts the compiler:

```
cc -qtr=r0 sample.c
```

The options used are -q, for the Quiet mode; -r, to make a linkable module; and -t=r0, which tells the C compiler to use the

```
#
# Sample Make file for test.c
#
Sample: sample.r
l68 /dd/defs/cstart.r sample.r /dd/lib/clib.l\
/dd/lib/clibn.1 /dd/lib/sys.1

attr sample -pee sample

#
# sample program compile
#
sample.r: sample.c
cc -qtr=r0 sample.c
```

Figure 1: Sample makefile

RAM disk for temporary files.

The RAM disk is included in the MODULES directory on my system. If you initialize /r0 (by entering in iz r0), OSK sets up a 425K RAM drive. This alone greatly speeded up my C compiling 100 times. Before I started using /r0, it would take anywhere from 5 to 10 minutes to compile

a simple program. Now it takes only about a minute to compile the same program. I understand that once I get the I/O board (with an additional two megabytes of memory) from IMS, compiling gets even faster. [Editor's Note: The author wrote this article prior to receiving the I/O board for his MM/1. Soon after our decision to




## Burke & Burke

P.O. Box 733 Maple Valley, WA 98038  
U.S. ORDER DESK: (800) 237-2409  
INT'L & TECHNICAL: (206) 432-1814

**Boost your CoCo with these fine Burke & Burke products:**

<b>THEXDER:OS9 -- NEW FOR OS9. Use your TANDY™ Thexder cartridge under OS9. By Alan DeKok.</b>	\$29.95
<b>The 6309 Book -- 6309 programming book by Chris Burke. Includes XSM assembler, disassembler, and DEBUG patches for OS9 Level 2.</b>	\$24.95
<b>PowerBoost -- 2 MHz enhanced HD63B09E processor w/ OS9 kernel and I/O patches (10% - 50% speed improvement). Note: soldering required for installation.</b>	\$29.95
<b>WORLD CLASS CHESS* -- Use Cyrus Chess cartridge w/ L2 OS9</b>	\$29.95
<b>FILE SYSTEM REPACK 1.1 -- Faster OS9 disk defragmenter</b>	\$29.95
<b>FILE RECOVERY SYSTEM -- Helps recover files from OS9 disks.</b>	\$24.95
<b>R. S. B. * -- Disk BASIC for Level 2 (BASIC ROM required).</b>	\$39.95
<b>EZGEN 1.10 -- EVEN FASTER! Handy &amp; powerful OS9 bootfile editor</b>	\$19.95
<b>WILD &amp; MV -- Use wildcards with OS9 commands; move files</b>	\$19.95
<b>PERTASCII -- Challenging OS9 game to make words from a list of random letters. Play against the computer, multi-user, or BBS.</b>	\$19.95
<b>ZCLOCK - Continuous time / date display on Level 2 screen</b>	\$9.95
<b>COCO XT -- Use PC MFM or RLL hard drives with CoCo! OS9 SW included (add \$30 for COCO XT-RTC version with real-time clock; add \$20 for XT-ROM hard disk auto-boot ROM).</b>	\$69.95
<b>DAGGORPATCH -- Transfers TANDY™ Dungeons of Daggorath cartridge to DISK BASIC. Adds disk I/O, screen dump, repeat.</b>	\$9.95

WA RESIDENTS ADD 8.2% SALES TAX. MasterCard & VISA accepted. U.S. COD's add \$3.75. Min. U.S. shipping \$4.00. Min. to Canada \$5.00. Please allow 2 weeks for delivery. Overnight or 2nd-day available for in-stock items. Software upgrades \$5.00 each w/receipt including U.S. shipping.  
Call or write for a free catalog of more exciting Color Computer products!



### SOLITAIRE CARD GAMES

**DuoDeck:** Double Deck Solitaires.  
Play Sly Fox or Windmill - both games of skill. \$19.95


**Classic Solitaire:** Klondike, Canfield and Pyramid on one disk for just \$14.95

**La Belle Lucie:** A true game of skill. \$14.95

All three play on 128K CoCo3, with joystick and disk.  
**SAVE \$10.00. BUY ALL THREE FOR \$40.00**

---

**NEW CRAZY 8's**  
**CBVSA 8.2**



Play against your computer. Up to three opponents. All graphics, for all ages. CoCo3. \$14.95

---

### PACDUDE 3-D!

What did the folks at the Atlanta CocoFest have to say about PACDUDE 3-D?  
**Incredible!**

Find out why even hardcore OS9ers said "I have to have this program."

Author: Brian O'Neill

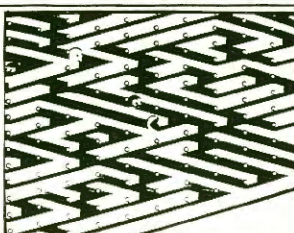
Runs on any 512K CoCo3. Requires joystick and disk drive. \$19.95

Checks, Money Orders, MasterCard & Visa \$2.50 S&H  
COD \$2.00 extra. WA addresses add 8.2% sales tax.

**(206) 653-5263**

**Eversoft Games, Ltd**  
P.O. Box 3354  
Arlington, WA 98223

30 day money back guarantee



---

**ADVENTURE SURVIVORS!**  
This is your newsletter! Only \$6 per year. Reviews, solutions, and more!  
Call or write for details.

L.E. Padgett  
24 Perthshire Dr.  
Peachtree City, GA 30269  
(404) 487-8461

Type	Colors	Graphics	Text	Border
0	16	640x210	80x26	Y
2	16	640x420	80x52	Y
3	256	320x210	40x26	Y
5	256	320x420	40x52	Y
6	16	640x210	80x30	N
7	16	640x420	90x60	N
8	256	320x210	46x28	N
9	256	320x420	46x60	N

Figure 2: KWindows Window Types

publish the article was made, he received the board. For his initial experiences with a complete system, see the sidebar "The I/O Board Arrives."

Another feature of /r0 is that you do not have to format it. Once you enter `intz r0`, the RAM drive is ready to go. On a 1-Meg system with the RAM disk in operation, you'll have 267K of free memory, which is still more than enough memory for most jobs.

Also included with the MM/1 is *Microware Basic*, a version of BASIC that is almost identical to BASIC09 under Level I or II on the CoCo. The difference is that *Microware Basic* does not include graphics support — *Microware* does not include a `gfx` or `gfx2` module. You do, however, get `syscall`, `inkey` and `rmb`. And I understand a `bcgfx` module (for *KWindows*) is in the works.

One of the things I like about *Microware Basic* is that, because of the way OSK works, I can use far more memory with it than with BASIC09. Under OS-9 Level II, the maximum memory size I could assign to BASIC09 is 42K. With *Microware Basic* and OSK, I can easily start with a workspace size of, say, 100K. Given the I/O board with additional memory, I could expand this to a megabyte and still have plenty of room for loading machine-code modules outside that workspace. If you use BASIC09, you'll have no trouble using *Microware Basic*.

The MM/1 comes with the *KWindows* windowing system. Because this system is

designed to make it easier for OS-9 Level II users to switch, the graphic primitives and escape sequences are almost identical to those in Level II. I signed onto a RiBBS bulletin board, and it opened and closed an overlay window on my MM/1 without any trouble.

Of course there are some differences, one of which involves screen types and resolutions. Using escape sequences or the `cgfx.1` library, you can create eight different types of windows with *KWindows* (see Figure 2). The VSC chip in the MM/1 supports a 720-by-560-pixel screen, but I understand that such modes are accessible only through assembly language. The higher-resolution graphic modes shown in Figure 2 are interlaced, a process by which multiple scans are used to create one display.

The thrust of the MM/1 is toward CD-I, which uses graphics images in the .IFF format. An .IFF viewer is included with the MM/1, along with a .GIF to .IFF converter. .IFF files provide a higher resolution than .GIF; many of the .IFF files I have look like photographs when displayed. Another type of graphic file supported by the MM/1 are .FL1 (flicker) animation files.

There are many advantages to OS-9/68000. The larger memory map means you don't have to worry near as much about running out of memory. The higher clock speed (16 MHz) allows your programs to run faster and paves the way for excellent graphics and animation effects. The SCSI

hard-disk controller is extremely fast. The higher screen resolution and greater number of colors allows more flexibility in your graphics.

On the software side, since *Microware Basic* is very similar to BASIC09, it should be easy to port any BASIC09 programs you have. Because it was designed with portability in mind, most C programs from OS-9 Level II should present no problem either. And C for OSK is very similar to C on Unix systems, making it a relatively simple matter to port most of that software.

OS-9/68000 V2.4 is very popular in Europe and Asia. For this reason, many programs have been ported to OSK. A series called TOPS is available from Europe and includes applications, utilities, games and even another Shell. Most of this software includes source code, so bug fixes and enhancements can be made.

Finally, I am already seeing many useful programs ported to OSK. Two of these are *CS* and *RCS*. The first is a popular Unix spreadsheet, and the second is a revision-tracking system many programmers use to keep track of the modifications, bug fixes and enhancements they make to programs. Also available is another C compiler as well as a C++ compiler.

If you are contemplating a move to OS-9/68000, make sure you consider the various systems that are available: the MM/1 from Interactive Media Systems, the System IV from Delmar and the TC-70 from Frank Hogg Labs. They all use OS-9/68000 V2.4, but for the most part, the similarity stops there. Each system uses its own internal bus, giving it certain advantages. For more information about the MM/1 and the System IV, see the reviews in THE RAINBOW (December and September 1991 issues, respectively).

The final decision is yours. Hopefully my I/O board will arrive soon so I can get my hard disk up an running. Then I can really begin taking advantage of OSK's power and the programs that are just waiting to be used.

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. John's Delphi username is VAXELF.

## The I/O Board Arrives

After I had written and submitted to THE RAINBOW my experiences with OSK and the MM/1, the I/O board I needed to complete my system arrived. Before the following weekend was out, I had installed an additional two megabytes of memory and a 130-Meg Maxtor SCSI hard drive.

The MM/1 manual caused some confusion regarding just how to install the I/O board; how you should orient the board in order to set the jumpers correctly was a little unclear. After a few unsuccessful attempts, however, I was able to get the jumpers in the right place, and the MM/1 booted fine with the 3-Meg Boot Disk supplied by IMS.

Once I was online with three megabytes, I created a new `OS9Boot` file using the New Module Upgrade from IMS. After this, I added the hard-disk modules (`hd` and `dd_h0`) to the bootfile. (For the sake of safety, I decided to exclude the `dd_h0` descriptor until I was sure the drive would format and operate properly.)

To format the hard drive, I first used the `attr` command to set the execute attribute of the `h0fmt` module and loaded that module into memory. Then I entered `format/h0fmt`. The SCSI interface used by the MM/1 is intelligent — the system determined the drive parameters by checking the drive before starting the format. When finished, I had a clean hard drive with just over 127 megabytes of free space. Once formatted, it was a simple matter to create appropriate directories (`CMDS`, `SYS`, etc.) and copy the files from the 3½-inch disks that came with the MM/1. Finally, I created a new boot disk, replacing the `dd_d0` module with `dd_h0`.

The stock EPROMs on the MM/1 processor board include code to support hard drives but are not capable of booting the system directly from a hard drive. Thus, you need to use a floppy boot disk just as with OS-9 Level II hard-drive systems on the CoCo 3. (IMS does offer a replacement EPROM set that first looks to the floppy drive for a boot disk. If no floppy boot disk is present, the code in these EPROMs tries to boot the system from the hard drive.)

With the I/O board installed, it was time to play with the other features it includes: stereo sound, mouse support and a printer port. The MM/1 sound capabilities are impressive. The port can be used to drive a pair of mini-speakers and provides enough oomph that, in most cases, you won't need to add an amplifier. The set of speakers I use has a built-in amp but no volume control, so I leave the amp turned off. (I do plan to add a volume control in the near future, though.) The sound-input side works equally well, but it does require a fairly large input voltage — it won't work if you simply connect a microphone to the port.

The original and updated mouse drivers are hard coded to work with the /L2 serial port. These drivers are designed for use with a Logitech C7-type (or compatible) mouse, although IMS now offers a mouse driver that supports the Microsoft serial mouse. I like the three-button Logitech mouse and recommend it for other users. As a side note, the newer version of *KWindows* supports a keyboard mouse that uses the numeric keypad.

The printer port (/p) works very well, and IMS is working on a modification to allow use of the second included port. To help those who have printers without a DIP switch to control linefeed settings, there is a jumper on the I/O board that handles this.

Now that I have a complete system, I am very pleased with the MM/1. Its speed is very impressive, and programs load and execute very quickly. And I like the way OS-9/68000 works. Best of all, I just learned that the TC-70 from Frank Hogg Labs will soon support *KWindows*. This means even more software for all of us OSK users.

### Feature Program

## What's in a BYTE? by Keiran Kenny

A byte is one of those odd little things computers use. It is a physical entity — a group of eight binary digits; electrical impulses — yet you'll probably never actually see one. Of course you can see the character represented by those eight bits that form a byte. And this is really all that's important.

*Bytes* is a BASIC program I wrote to illustrate what bit combinations are used to represent different ASCII characters. When you run the program, you see a quick overview of how a byte works, then you are asked to enter values between 0 and 255. *Bytes* shows you the actual character for the value along with its binary value. If you press N when asked if you want to enter another value, you are prompted to press keys on the CoCo keyboard, for which *Bytes* displays the proper values.

Keiran Kenny's interests lie mainly with the Color Computer's graphics and math capabilities. But in his own words, "I like to try everything." He may be contacted at vanMontfoortlaan 31, 2596 SP The Hague, Holland.

16K

### The Listing: BYTES

```

1 'BYTES
2 'BY KEIRAN KENNY
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 CLS
20 PRINT@42,"***BYTES***"
30 PRINT@193,"BY KEIRAN KENNY, S
YONEY, 1989."
```



```

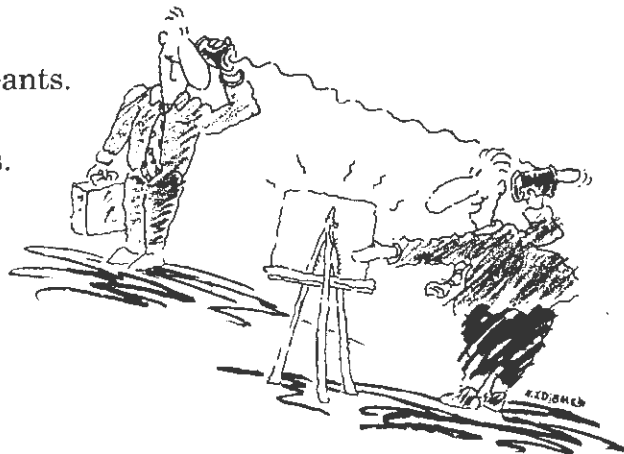
40 GOSUB340
50 CLS:PRINT@96,"A BYTE IS LIKE
A ROW OF EIGHT SPOTLIGHTS. EAC
H LIGHT IS A BIT.WHEN ALL ARE ON
THEY REPRESENT 255 AND THE BIN
ARY NUMBER...":
60 X=294:C=256:GOSUB260:GOSUB340
70 CLS
80 PRINT@128,"WHEN ALL LIGHTS AR
E OFF THEY REPRESENT ZERO AND
THE BINARY NUMBER...":X=294
:C=0:GOSUB260:GOSUB340
90 CLS:PRINT@32,"IN VARIOUS COMB
INATIONS OF ON AND OFF THEY RE
PRESENT THE CHAR-ACTERS ON YOUR
KEYBOARD AND MANYOTHER CHARACTER
S. FOR INSTANCE THE LETTER 'A'
IS DECIMAL 65 ANDBINARY...":X=2
94:C=66:GOSUB260:GOSUB340
100 CLS:PRINT@32,"INPUT A DECIMA
L NUMBER IN THE RANGE 0-255":
INPUT
110 IFT>255THENCLS:PRINT@32,"MAX
IMUM 255. TRY AGAIN.":FORU=1:1010
00:NEXT:GOTO100
120 TS=RIGHT$(STR$(T),LEN(STR$(T
))-1):D$="CHR$(T+T$)/CHR$(&H+
HEX$(T)+")"
130 IFT<330RT=143THENPRINT@128,0
$+" NOT VISIBLE"ELSEPRINT@131,0$
+" "+CHR$(T)
140 PRINT@196,"BINARY "+T$+" -":
X=294:IFT=0THENC=T:GOSUB260:ELS
EC=T+1:GOSUB260
150 GOSUB330
160 IFK$="Y"ORK$="y"THEN100
170 IFK$="N"ORK$="n"THENCLS:GOTO
190
180 GOTO150
190 PRINT"PRESS ANY KEY ON YOUR
KEYBOARD EXCEPT <BREAK>. USE TH
E <SHIFT> KEY AND LOWER CASE TOO
.":GOSUB350
200 TS=RIGHT$(STR$(ASC(K$)),LEN(
STR$(ASC(K$))-1):IFASC(K$)<330R
ASC(K$)>122THENPRINT@131,"KEY":E
LSEPRINT@131,K$;
210 PRINT" = CHR$(T+T$)/CHR$(&
H+HEX$(ASC(K$)+")"
220 PRINT@195,"BINARY "+T$+" -":
X=294:C=ASC(K$)+1:GOSUB260
230 PRINT@354,"NOW ANOTHER KEY O
R <BREAK> TO END PROGRAM."
240 GOSUB350
250 IFK$<>CHR$(3)THEN200ELSEEND
260 FORP=7:TOSTEP-1
270 IFC=(2^P)THENPRINT@X,CHR$(1
59):PRINT@X-33,1;
280 IFC<<2^P)THENPRINT@X,CHR$(12
8):PRINT@X-33,0:GOTO300
290 C=C-(2^P)
300 X=X+2
310 NEXTP
320 RETURN
330 PRINT@424,"ANOTHER? Y/N":GOT
0350
340 PRINT@424,"PRESS ANY KEY":
350 K$=INKEY$:IFK$=""THEN350ELSE
RETURN
```



# 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.



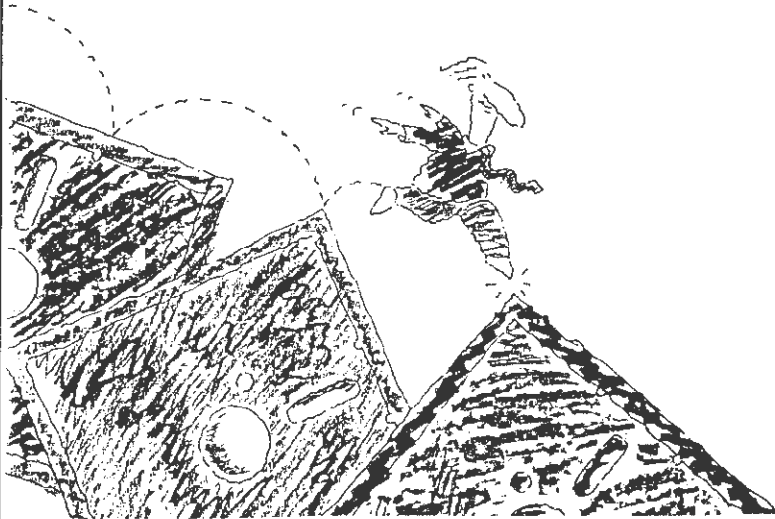
## Trial Offer: 5 hours for \$5!

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



# PCM

The Premier Personal Computer Magazine for Tandy Computer Users

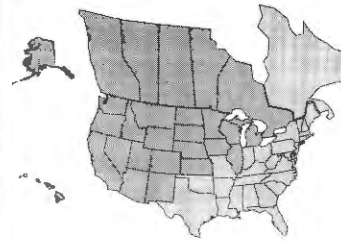
Covering the Complete Line of Popular Tandy MS-DOS and Portable Computers

Call (502) 228-4492 for information: (800) 847-0309 for VISA/MC/AMEX orders  
PCM is a registered trademark of Falsoft, Inc.

## Advertisers Index

Burke & Burke .....	15	Owl-Ware .....	7
Computer Plus .....	BC	PCM .....	19
Dayton Associates of W.R. Hall, Inc. ....	13	Rainbow Back Issues .....	18
Delphi .....	17	Sundog Systems .....	5
Eversoft .....	15	Sundog Systems .....	19
Granite .....	9		
Hawksoft .....	19		
Infinitum Technology .....	9		

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



Call Kim Lewis  
Eastern Sales Director  
(502)228-4492

Call Ira Barsky  
Western Sales Director  
(312)587-1818

The Falsoft Building  
9509 U.S. Highway 42, P. O. Box 385, Prospect, KY 40059  
(502) 228-4492 • FAX (502) 228-5121

## The Contrás

2 years in the making.  
A cast of 3 all-star programmers.  
THE game. Here at last...

Once upon a time, Sundog Systems announced a new 512k game called *The Contrás*. It took two years and three programmers to complete this, the most ambitious game ever created for the CoCo-3. Jeff Steidl, accomplished author of *Photon* and *GrafExpress 2.0*, led the effort to produce this technological marvel. In doing so, he proved that the CoCo can match—or surpass—any home game system.

*The Contrás* features a two-player cooperative mode, 512k filled with incredible graphics, snper-smooth animation and scrolling, an outstanding background music score, sizzling sound effects, and lightning-fast arcade action.

This is paramilitary combat at its best. Play alone or with a friend as you take on the evil alien invaders. Blow away the enemy while travelling thru multiple levels and powering up with ever more destructive weapons. *The Contrás* will keep you playing for hours; it is quite possibly the best CoCo game ever! Requires 512k CoCo-3, disk drive, & joystick.

**\$34.95**

## Super Sundog Sale!

For this holiday season, Sundog is cutting prices on even our most popular products. Give more to your favorite CoCo gamer for less! You must mention this ad when ordering to get these very special prices! See our full-page ad elsewhere in this issue for prices & shipping info.

### 10% off 2 or more

- GrafExpress 2.0
- Photon
- Kyum-Gai (either ver.)
- Zenix
- Crystal City
- Sinistaar
- Quest for Thelda

### 10% off

- War Monger
- Soundtrax
- Warrior King
- Quest/Starlord
- All hint books

### 50% off

- Hall/King 1,2, or 3
- Paladin's Legacy
- Kung-Fu Dude
- Champion
- Dragon Blade
- White Fire/Eternity



Announcing

## Icon Basic09

The next programming language for OS-9!

**Icon Basic09** is a graphical user interface (GUI) to Basic09, which will make programming easier than ever! **Icon Basic09** takes an innovative approach by using graphic representations, or icons, to represent statements and keywords for writing Basic09 programs and procedures. Instead of constantly typing while writing a program, the user can simply **point & click** to choose the desired statement!

**Icon Basic09** can also be very useful in studying procedures and programs written by others to learn how they operate. The package contains a full set of icons...or, you may edit or create icons using the included icon editor. **Icon Basic09** requires a CoCo-3 with at least 256k, mouse or joystick, and OS-9 lv 2.

**\$20**

- Dual hi-res joystick adapter** (RS/Colorware) **\$40**
- Hi & Lo-res joystick adapter** **\$27**
- HAWKsoft keyboard extension cable** **\$25**
- Demination** ("Risk"-like wargame!) **\$18**
- MyDOS** full-featured DOS extension **\$15**



PO Box 7112  
Elgin, IL 60121-7112  
(708) 742-3084 eves & ends

US and CDN S&H always included. Terms: MO, check, or COD.

From Computer Plus to YOU . . .  
**PLUS after PLUS after PLUS**



**Tandy 1110 HD \$739\***  
**Tandy 3800 HD \$1559\***  
**Tandy 3830 SL \$1749\***



**Tandy 2500 RSXHD \$749**  
**Tandy 4825 SX/25 HD \$1279\***



**Tandy 4850 EP \$1749\***



**DMP-135 \$189\***



**Color Computer Disk Drive**  
**Drive 0 \$239 Drive 1 \$149**



**TandyFax 1020 \$299\***

**BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS**

**COMPUTERS**

Tandy 1000 RSX FD, VGM 225	559.00 *
Tandy 1110 HD 1 Drive 640K	739.00 *
Tandy 3800 HD 1 Drive	1559.00 *
Tandy 2500 RSXHD 1 Meg RAM	749.00
Tandy 3830 SL 1 Drive 2 Meg RAM	1749.00 *

**PRINTERS**

Tandy DMP-136 213 CPS	199.00 *
Tandy DMP-302 270 CPS	469.00
Tandy DMP-202 180 CPS	299.00
Tandy DMP-442 300 CPS	539.00
Tandy LP-950 Laser Printer	1299.00
Tandy DMP-240 192 CPS 8 color	329.00 *
Panasonic KXP 1180 192 CPS	189.00 *
Panasonic KXP 1123 24 Wirehead	259.00 *
Panasonic KXP 1124 i300 CPS	329.00
Okidata 320 300 CPS	369.00
Okidata 380 180 CPS 24 Wire HD	239.00
OKI Laser 400 4PPM	679.00 *

**MODEMS**

Tandy DCM-6	52.00
Tandy DCM-7	85.00
Cardinal 1200 Baud External	99.00
Cardinal 2400 Baud External	129.00

**COLOR COMPUTER MISC.**

Tandy Drive Controller	89.00 *
Extended Basic Rom Kit (28 pin)	19.95
64K Ram Upgrade Kit (2 or 8 chip)	39.00
Tandy Deluxe Keyboard Kit	24.95
HI-RES Joystick Interface	8.95
Color Computer Deluxe Mouse	44.00
Multi Pak Pal Chip for COCO 3	14.95
COCO 3 Service Manual	29.95
Serial to Parallel Converter	59.95
Tandy Deluxe Joystick	19.95
Magnavox 8135 RGB Monitor	299.00
Magnavox Green or Amber Monitor	99.00
CoCo 3 Game Chip	29.95
Tandy Pistol Grip Joystick	26.95
PBJ OK COCO 3 Upgrade Board	29.95
PBJ 512K COCO 3 Upgrade	89.00
Tandy OK COCO 3 Upgrade Board	39.95
Tandy 512K COCO 3 Upgrade	99.00 *

**COLOR COMPUTER SOFTWARE**

	TAPE	DISK
The Wild West (COCO 3)		25.95
Worlds of Flight	34.95	34.95
Mustang P-51 Flight Simul.	34.95	34.95
Flight 16 Flight Simul.	34.95	34.95

Tandy Educational Software	2.00
Spinnaker Software	2.00
Max 10 by Colorware	79.95
AutoTerm by PXE Computing	29.95 39.95
TW-80 by Spectrum (COCO 3)	39.95
TeleWriter 64	49.95 59.95
TeleWriter 128	79.95
Elite Word 80	79.95
Elite Calc 3.0	69.95
CoCo 3 512K Super Ram Disk	19.95
Home Publishing by Tandy (CoCo 3)	35.95
Sub Battle Sim. by Epyx (CoCo 3)	26.95
Thexder by Sierra (CoCo 3)	22.45
Kings Quest III by Sierra (CoCo 3)	31.45
Flight Sim. II by SubLogic (CoCo 3)	31.45
OS-9 Level II by Tandy	71.95
OS-9 Development System	89.95
Multi-View by Tandy	44.95
VIP Writer (disk only)	69.95
VIP Integrated Library (disk)	149.95

Prices are subject to change without notice. Please call for shipping charges. Prices in our retail store may be higher. Send for complete catalog.

\*Sale prices through 12-10-92

**CALL TOLL FREE**  
**1-800-343-8124**

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



**computer plus**

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

**SINCE 1973**

**IN MASSACHUSETTS CALL (508) 486 3193**