

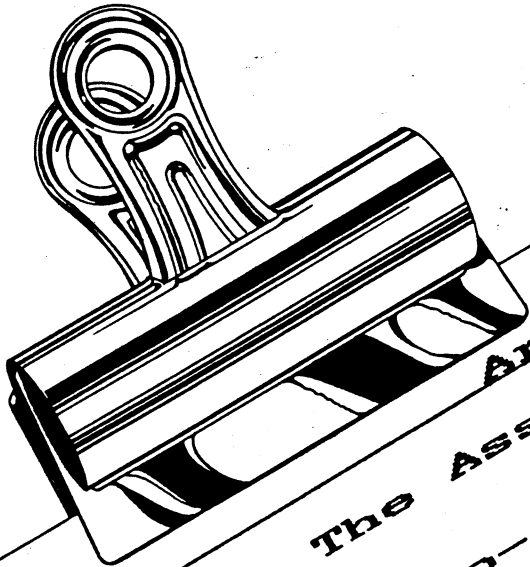
Vol. 1 # 1 Sept./Oct. 1987

\$2.50 U.S.

\$3.50 Can.

CoCo

Clipboard Magazine



- Writer Article
- X. Brockschmidt... The Assembly Line
- R. Caley... CoCo-izing Your Office
- W. Day... On-Line with Compuserve
- J. DeStafeno... A Hard Drive For You?
- M. Dooley... CoCo & Amateur Radio
- R. Krippner... "The 9th. Power"
- D. Searles... Pascal Programming
- B. van der Poel... BASIC 09 / HDIR

C'mon - Join The Adventure!

ORDER PHONE (416) 456-0032

Duck Productions, 18 Rowe Court, Brampton, Ontario, Canada L6X 2S2
Please add \$2.00 for handling. Ontario residents add 7% provincial tax

DUCK QUALITY COMPUTER PRODUCTS PRODUCTIONS



MACHINE GENESIS the teacher.

This package is a program and tutorial designed in a simple plain language fashion. It is a primer for the beginning assembly language programmer. A simplified learning tool, providing a solid background in the fundamentals of Machine language. Machine Genesis is full of knowledge and understanding that seems missing in little more advanced tutorials. In a step by step tour you will find a clear understanding and along the way, the you will learn a great deal about how your computer works. The disk programs included are:

- a full function Machine Language Assembler
 - a complete de-bugging system
 - a dis-assembler for M.L. examination
 - a 'Peeker' program for memory examination
 - a file input system for quick routine creation
- The manual provides very complete chapters on:
- hi-res graphic screen dumps
 - full printer control
 - Error trapping disk access from Basic
 - Controlling disk functions with Machine Language
 - File buffers, operation and applications
 - Cassette file functions and control
 - Text and graphic screen manipulation
 - The complete memory map
 - Keyboard access and uses
 - Joystick Control

and much, much more!

REQUIRES: 64K, Disk drive, Joystick, Dot Matrix or Daisy Wheel printer.

All For Only \$31.95 (\$39.95 CDN.)

MAP'N ZAP semi automatic disk repair

This is a package of programs with simple instructions, designed to be the layman's kit for disk directory repair. Another plain language solution for easy application. The manual provides a full, disk system tutorial explaining Coco's input/output operations. A simple walk through program uses and DISK BASIC I/O methods, finishing up with an easy step by step procedure for disk directory repair. With Map'n Zap, your Coco will list the disk contents that are identifiable to your printer or screen, helping you locate directory errors. For the most severe cases, the included copy utility is designed to read and write any errors to a fresh disk for repair. It is error trapped and fully prompted to insure ease of use.

REQUIRES: 32K, disk drive.

This complete package \$19.95 (\$24.95 CDN)

CODE BUSTER code disassembler.

This is an accurate and dependable method of disassembling and examining any machine language program including Coco's BASIC ROMS. The program's purpose is as a tool to investigate how other programmers have been writing various routines or to get an understanding of Coco's Disk Basic routines to aid you in your own programming. This basic utility will provide screen or hardcopy output, the printed version being complete with decimal equivalents. Also included on this disk is:

'ADDRESS' will quickly and easily provide start, end and execution address's and the length of any binary file.

'ROM RAM' rewrites the BASIC ROM to RAM memory permitting you to experiment with making changes to BASIC's subroutines.

'MINIPEEK' allows examination of memory contents and comparison to their ASCII counterparts.

We've included heaps of specific information on the three Roms, their layout and subroutine locations. An affordable adventure in learning.

REQUIRES: 64K, Disk Drive, Printer

Specially priced \$19.95 (\$24.95 CDN)

TRIED & TRUE tasty recipes

We can all agree there is only one pastime that may be better than computing. Yes, in those moments between programs we have all raised our thoughts to food. In grand style, we have brought the two together in an attempt to show that computer pallor can be a healthy complexion. This disk contains over one hundred recipes to dazzle your tastebuds. It features a wide and varied array of food types, in a range from meals to beverages. Also, included is a simple disk reader for those of you who do not have a word processor.

Over one hundred bargains at \$14.95 (\$19.95 CDN)

LASER MAZER master puzzle of reflection

Would Mr. Sulu shoot himself in the aft with a wayward phaser blast? Would Spock learn 3 dimensional chess before mastering theories of relative movement? Would Captain Kirk give the wrong co-ordinates to zap the Klingons to Tribble land? No!No!No!They all learned the hottest game at Starfleet. LAZER MAZER! Wait! Whats this? There are six cloaked Romulan vessels entering your quadrant and preparing to attack Regula One. Where are they? Can you find and engage them before they reduce the scientific research laboratory to a good idea. Ah! but the Captain has an edge. Fire selective sublight frequency, low light laser blasts into space and monitor the results with ships sensors to find them. The Romulan will never know what gave them away. Then again, maybe your Laser Mazer skills are not what they used to be! Maybe the Romulans will get You!

Special Sale \$14.95 U.S. (\$19.95 CDN)

CLASS MONITOR dual monitor driver

This monitor driver circuit designed for use with any Coco, will drive any composite video monitor. It provides separate monochrome and color outputs, as well as line level audio output for immediate access of either mode. This valuable addition to your computer does not require soldering, nor will it affect the RF TV output currently in use. Step by step installation instructions are easy for anybody to follow or you may decide to call the TV repair shop for the handymans touch. Write to us for further information. Assembled Class Monitor and documentation

Total price \$31.50 (\$39.50 CDN)

CLASS MONITOR KIT

Now you can build your own monitor driver and save money. This kit comes complete with all the instructions and information necessary to construct your own video driver. Class Monitor - The Kit \$9.95 (\$11.95 CDN.)

MICRO • FIRE the ultimate secret weapon

Had a tough day defending the Earth?

Tired of beating your thumbs more often than the aliens?

Then this is just the circuit for you. Micro•Fire is compatible with any joystick or computer system and gives you automatic rapid fire by merely holding your fire button. Micro•Fire is an easy installation for the layman. It has a few hidden options to tailor your system just the way you like it. You may install and use Micro•Fire two different ways. You may attach Micro•Fire to your existing fire button with the switch supplied. This will give you on / off (rapid fire / single shot) control. Micro•Fire also includes all wires and notes necessary to install with a second fire button, or on the new two button joysticks, giving instant choice of weapons.

Complete for \$19.95 (\$24.95 CDN)



MICRO • FIRE KIT

Now you can build your own rapid fire system and save money. This kit comes complete with all the instructions and information necessary to construct your own Micro•Fire.

Micro•Fire - the kit \$7.95 (\$9.95 CDN.)

DUCK SOUP NO. 1 bargain disk'o plenty

A bargain potpourri of programs, hints, and tips. At least fifteen unusually useful utilities, games or interesting projects to stimulate your Coco fever.

\$12.95 (\$16.95 CDN) (That low price is not a misprint)

MAKING TRACKS disk management.

MAKING TRACKS is a nine program disk that goes beyond user friendly to become user independent. Its main purpose is to organise your Coco and disks without the burden of boring time and keystrokes, but you will find a whole lot more.

'D' is a high-speed directory boot that will give you continuous access to any length of directory without retyping DIR, and it will autostart any disk program with a single keystroke. The 'D' program is put on each disk. When you type RUN'D the disk title appears, providing the disk name, number, drive number, the free space, the total number of files (on and off the screen), as well as the (scrollable) directory that includes the size (number of grains) of each program. A floating cursor awaits arrow commands to choose and autostart any program (basic or binary).

The 'D' Creator - Creates the 'D' binary file, prompts you for the disk name identification and installs the finished product on any formatted disk consuming only 1123 bytes or 0.7 % of 161,280 bytes of disk space.

DOS Creator - Creates the same as above but a 'directory invisible' version of 'D' that is accessed by typing the command DOS instead of RUN'D

ALPHA DIRECTORY - will sort your directory and rewrite it alphabetically. This is for easy, single keystroke examination of many directories, one after another as when searching for a particular file.

ADDRESS FINDER - will find address's of machine language programs as well as the execution entry and the total length.

DISK COMPARISON - will keep your backup disks current without any labor or typing from you. The program compares the directories of two disks and reports those files that are not duplicated. By single keystroke command, you choose which files will be automatically transferred to the appropriate disk. The selections are copied in succession like a backup.

DIRECTORY PRINT - creates a printed catalogue in 8 1/2" x 11" format, of all your disk's contents. With single keystrokes, Directory Print will access the 'D' program on each disk to read the disk title, number or identification and amount of free space as well as the directory contents. All of this information is organized for fast printer output in two dual columns.

DIRECTORY SAVER - a single keystroke provides an insurance policy against program loss by reading and copying the important grain table sectors to a safe area, high in the directory track. Similarly, when disaster strikes, you can retrieve this information and repair your directory in seconds with a single keystroke.

DRIVE SPEED TEST - will provide a method to detect disk problems before they get out of hand. Readouts are provided to assist in speed adjustments.

SET PARAMETERS - to maintain the file for your personal system specifications. This infofile is accessed by all MAKING TRACKS programs to operate within your predefined requirements.

Complete documentation and full startup methods are included.

REQUIRES: 32K, Disk Drive, Printer

All this for only \$19.95 (\$24.95 CDN)

The depraved programmers chained to their terminals on sublevel 18 are hoping to hear from you. If you have any questions, suggestions or God forbid, complaints, we want to hear them. We are very interested in your fresh ideas for software, as your wish could be our command. We are genuinely interested in your input, raw or finished, amateur and expert alike.

Write to us at: Duck Productions
18 Rowe Court
Brampton, Ontario Canada
L6X 2S2
Order Phone: (416) 456-0032

(When writing please include your area code and phone number)

Buy Quality and Value!

CoCo Clipboard Magazine:

CoCo Clipboard Magazine

Vol. 1 Issue 1 Sept./Oct. 1987

TABLE OF CONTENTS

HDIR
by Bob van der Poel page 4

"The 9th Power"
by Randy Krippner page 5

Pascal Programming
by Delmar Searles page 8

"The Assembly Line"
by Kraig Brockschmidt page 11

On Line with CompuServe
by Wayne Day page 14

CoCo 'N Amateur Radio
by Mike Dooley page 17

Is a Hard Disk in Your Future?
by Jim DeStafeno page 20

CoCo-izing Your Business
by Rush T. Caley page 25

* * * * *

Subscription Information page 15

CoCo Clipboard Coupons page 16

Sept. / Oct. 1987 Issue 1

Editor and Publisher
Theodore W. Paul, jr.

Associate Editor
Darlene M. Paul

Contributing Writers:
Rush T. Caley, Jim DeStefano, Kraig
Brockschmidt, R. Wayne Day, Mike
Dooley, Bob van der Poel, Delmar
Searles, Randy Krippner

Production & Printing by:
Ron & Debbie Pacos (and friends)
Arkwright Printing Co., Inc.
Fredonia, New York 14063

CoCo Clipboard Magazine™
is published bi-monthly (6 times a
year) by CoCo Clipboard Magazine
3742 U.S. 20, Box 3
Fredonia, N.Y. 14063
(716) 679-0126

ENTIRE CONTENTS COPYRIGHTED

NO PORTION OF THIS MAGAZINE MAY BE
REPRODUCED, BY ANY MEANS, EXCEPT BY
EXPRESS WRITTEN PERMISSION OF THE
PUBLISHER.

Tandy, Extended Color Basic, Color
Basic and Program Pak are
registered trademarks of Tandy
Corp. CompuServe is the registered
trademark of CompuServe, Inc.

All programs, information and
articles are distributed on an "as
is" basis without any warranty of
any kind whatsoever.

CoCo Clipboard, The Clipboard,
Clipboard BBS Systems, "Clipper
Koala" and Clipboard Convo are
trademarks of CoCo Clipboard
Magazine.

Subscriptions to CoCo Clipboard
magazine are \$12.00 per year for 6
issues. We accept checks or money
orders only. Canadians may pay
\$12.00 in U.S. funds. All overseas
subscriptions are \$25.00 surface
mail.

Orders may be sent to:
CoCo Clipboard Magazine
3742 U.S. 20 Box 3
Fredonia, N.Y. 14063

Full refund after one issue. No
refunds after two issues.

Your comments and suggestions are
welcome. We reserve the right to
edit and publish all letters
received unless requested not to by
the writer.

Welcome to the first issue of CoCo Clipboard Magazine. You might have thought that the publisher would take a lot of space in the first issue to introduce the magazine, it's writers, it's philosophy, etc. etc. I hope you won't be disappointed if I don't.

You see our philosophy at CoCo Clipboard is "all meat, no cereal filler" regarding our articles, programs and tutorials. Frankly we're tired of the large doses of fluff passed off as useful information these days. Many of you have agreed with your subscriptions and ads.

And speaking of ads you'll find something here in Clipboard you'll find nowhere else - COUPONS! No they're not for laundry soap or hair spray, they are money off, discount or free sample coupons provided by our advertisers to you. CoCo Clipboard will run money savings coupons in every issue. You'll save enough for your subscription to Clipboard and maybe a new CoCo 3 too!

Finally about our subscriptions. We publish every other month. A subscription for one year of six issues is \$12.00. Our friends in Canada can also pay \$12.00 in US funds for their subscriptions. For \$2.00 a month you'll get some of the finest programs, articles and tutorials available to the CoCo family. Your check or money order is welcome. To save on costs we do not accept any credit cards, but there is no 3 or 4 month delay in getting your subscription started either. Fill out the form here in the magazine or drop us a note with your check. We'll get your name on the CoCo in the office and your Clipboard in the mail. Your letters and comments are welcome. It's your magazine, tell us what you need and want. We'll respond as fast as we can.

We hope you enjoy this first issue and extend you the following invitation:

C'mon Join The Adventure!

HDIR

Bob van der Poel

Well, I finally got my copy of Level II OS-9 . . . complete with its 20 pound manual! I'd fooled around with BASIC09 back in the old Level I days, but with the memory restraints, etc. I was never all that impressed -- even though I did know that BASIC09 had some very powerful features. With 512K (this program will work fine in 128K systems) the reasons for not using OS-9 have vanished. And what better way to get learn about OS-9 file structures than to write an hierarchical directory program? If you don't know what that is, read on! And what better way to learn about BASIC09 than to use it?

As you'll be told by any OS-9 aficionada, one of the powerful features of OS-9 is its directory structure. Each disk has a "root directory." This directory can contain files (both "programs" and "data" are files) or other directories. The other directories can also have files or directories in them, and so on to as many levels of directory as you have room on the disk. By way of example your disk might have a directory called "CMDS" with all the executable programs in it and another directory called "TEXT.FILES" with wordprocessing files in it. Since the whole family uses your computer TEXT.FILES has three more directories in it: MOMS.FILES, MY.FILES and JUNIORS.FILES. Making things even more complex is that the directory MY.FILES has the LETTERS and SHORT.STORIES directories in it. Finally, LETTERS contains copies of your letters to Aunt Martha and your mom (you do write to them, don't you?)

The problem with all this organization is that it is very difficult to find out what is on a disk. After a couple of hours of doing CHDs and DIRs I decided that there had to be a better way. Was it possible to let the computer weave its way through the various levels of directories and print out

a complete listing of all the files and directories on a disk? And was this the project which would show off BASIC09? The answers to both these questions are yes -- if they weren't you wouldn't be reading this.

The BASIC09 program "HDIR" demonstrates a number of features other BASICs lack:

1. Modular programming: By writing programs in small, understandable chunks debugging is easier -- especially two months from now when the original program logic has been forgotten.
2. Meaningful variable names.
3. Complex data types.
4. Recursion: This is when a program module calls itself.
5. The ability to do calls direct to the operating system for low level tasks not directly supported by the BASIC09.
6. Structured code using loop constructs rather than GOSUBS and GOTOS.

With the comments contained in the programs, and lots of paging through the OS-9 manual, you should be able to figure out most of the program. I'll try to cover some of the more esoteric parts here:

The first system call in HDIR determines the device name of the standard input. This will be something like "TERM" or "W7," unless of course you have redirected input from another device which you must NOT do with this program. This device is opened for output -- this is the only way we can print a message on the screen if the standard output of program has been redirected. This means that after packing the program into the CMDS directory you can do a directory to the printer by typing "HDIR >P" or to a disk

file with "HDIR >/d1/directories/myname."

The second system call is done to find out the name of the device the directory you have input is on. If you typed "/D0/CMDS/basic09.programs" in response to the prompt the system call will return "D0", "/d1" will return "D1", etc. Once we know the name of the device the root directory is on, we can open that device (in this case a disk drive) for reading as one giant file. We need to read sectors on the disk (without regard to file structure) in order to determine the actual file structure of the disk. If this sounds complicated, it's probably because it is -- have a good read through chapter 5 of the Technical Reference section of the OS-9 manuals.

If the user has requested an HDIR of the entire disk (eg. "/D0") some stats are pulled off logical sector 2 and the disk name and creation date are printed; otherwise the name of the directory where HDIR starts is printed (eg. "/d1/programs"). With all this housekeeping out of the way the main program, HDIR1, is called.

HDIR1 scans through the directory in two passes. On the first it skips all directory entries in the directory and prints the names of files. If any directories were found, it makes a second pass through the directory -- this is where the recursion takes place.

If a directory entry has been found, the variable "direntry" is set to the name of the directory. The command "CHD direntry" sets the current directory to "direntry" and then the program calls itself. This new incarnation of the program

The 9th Power

Randy Krippner

Welcome to the Coco Clipboard and to the first installment of The 9th Power, a column about the OS9 operating system and the BASIC09 programming language. Since this is a new column in a new magazine, let me give you an idea of what I'd like to do.

First, this column is going to deal with the Coco 3 and OS9 Level 2. I realize that this leaves out a lot of Coco 1/2 owners who use OS9 Level 1, but there isn't enough space to cover both versions.

Second, most of the material published about OS9 L1 has been written with the techno-wizards in mind. You know who I mean, the people who balance their checkbooks in hexadecimal and who's idea of a fun Saturday night is hand assembling a 20K machine language program. In binary.

This may have been alright for OS9 L1 because it seemed that only the more technically oriented Coco owners used OS9 L1. But since OS9 Level 2 is the only real way to access the full capabilities of the Coco 3, and since in all likelihood a great deal of software for the CoCo 3 is going to run under OS9, non-technical people will be buying and trying to use OS L2 as well. So I'm going to try to make the discussion here as non-technical as possible.

Third, almost everyone I've talked to mentioned they were interested in writing their own programs. It seems that most readers of Coco magazines are programmers, at least on a hobby basis, so we're going to dabble in programming as well, using using BASIC09.

BASIC09 is a programming language that combines the best features of BASIC and Pascal. It has been largely ignored until now is because it originally cost \$100 extra for OS9 Level 1 and because

there was only about 14K of program space available on thCoCo 2. Tandy now includes BASIC09 with OS9 L2 at no extra cost and RAM is of no concern any longer on a 512K CoCo 3.

Fourth, I'm going to be skipping around a bit. OS9 is such a difficult subject to cover comprehensively that I'm not even going to try to explain every part of it. What I'm going to do is cover the more difficult or more interesting parts of the operating system, including those that seem to give people the most trouble.

Now, if this were the usual OS9 column the next thing you'd see would be a dull explanation of OS9's file structure and basic functions. But I'm not going to do that. I don't think it's necessary.

If we still had to deal with the original OS9 Level 1 manuals, it would be a different story. But Tandy did something right for a change. While the OS9 Level 2 manual is one of the most intimidating looking documents I've ever seen as far as sheer size is concerned, it is actually well written. The introductory chapters in the "Getting Started" section of the manual covers the basic material quite well. Still, because of the huge size of the manual, Tandy might have done well to follow the example of the Hitchiker's Guide to the Galaxy and printed "Don't Panic" on the front cover.

If people write and request that the more basic functions and concepts of OS9 be explained in more detail, we'll backtrack. But for the time being, we're going to jump right in.

There is one hint I'd like to pass along before we move on, however. DON'T USE THE CONFIG UTILITY IF YOU ONLY HAVE ONE DISK DRIVE! I tried it before I picked up a second drive for my Coco 3.

It took almost an hour to make a boot disk using CONIG, and I finally gave up trying to count the number of disk swaps it took when I reached 45.

Now, let's take a look at one of the most exciting parts of OS9 Level 2, windows.

"Windows" is the buzzword in computer circles these days. Window capabilities are one reason why the 68000 computers such as the Atari ST are popular. A windowing user interface from Microsoft is available for IBM PC clones (which Microsoft named in a remarkable burst of creativity, "Microsoft windows") and which, rumor has it, will be standard on the new IBM PS2 computers.

I'm going to assume you've read the Getting Started section of the OS9 manual and are familiar with the basic functions of OS9. I'm also going to assume you have a Coco 3 with 512K RAM. You can run OS9 with just 128K, but you start running out of memory fast when you begin play with windows as we are going to. Even with 128K, though, you should be able to run many of the examples given in this article.

First, what is a window? A window is an area of your screen set aside to display information other than that which appears on the main screen, sort of like a screen within a screen. Windows are used to display "pop-up" menus, tables, help information, etc. But under OS9 L2, windows are much more sophisticated than this.

There are actually two types of windows under OS9; device and overlay windows.

Continued next page

Device windows are unique to OS9. I haven't made a complete survey of the entire computer market, but I haven't encountered anything quite like them anywhere else, and they are very powerful.

OS9 is a multi-user, multi-tasking operating system. Multi-user means more than one person can use the computer at the same time by hooking a terminal up to it through an RS-232 port. Multi-tasking (also called multi-processing) means that OS9 can run more than one program at the same time.

Device windows take advantage of these capabilities. When a device window is created, you are actually creating a terminal, a second computer, so to speak, from which you can run programs, execute OS9 commands or do anything you could do from the main screen. Meanwhile, the original screen is still there, doing its thing at the same time. You could, for example, start printing a lengthy file from disk from one window and then jump into a second window to edit another text file at the same time.

Device windows

are unique

to OS 9

But nothing beats an example. Let's create a device window. (In the examples, whenever you see <ENTER> it means press the ENTER key.)

Boot up OS9 (use a copy of the original system disk, not one you may have customized for your own use). Then type in the following lines:

```
iniz w1<ENTER>
shell i=/w1& <ENTER>
```

After you enter the last line, you should see a number like &003 appear, then the OS9: prompt should reappear on the screen. The actual number you see may vary.

What you just did was initialize a device window, /w1, and fork a shell to it. The "/w1" is a device name.

Fork? Shell? No, I'm not talking about eating oysters. The shell is an OS9 program, like all of the other OS9 programs and commands you type from the keyboard. It works as sort of an interpreter, passing the commands you give it to the operating system for execution.

Fork is the term used to describe starting a program running from a different program, like a branch forking off from the main trunk of a tree. The second line in the above example starts an immortal shell running in /w1 while our original shell keeps operating on our current screen. The number you saw is the number of the process you just started. Whenever a new program is run it is assigned a process number.

To get to the new window, press CLEAR. CLEAR is used to toggle between device windows. You should see a small display in the upper left corner of your screen with an OS9: prompt in it. If we hadn't set up a shell in the window, there would be no OS9: prompt and it would be impossible to do anything except display information in the window. But because we activated a shell in /w1, this window is now that amounts to a second computer. Type any OS9 command; dir, mdir, mfree, etc. They will all work, although the display will look rather odd because of the small size of the window.

Press CLEAR to go back to the original screen. A shell is still active here as well and you can do anything you normally could with

```
deiniz w1 <ENTER>
```

This "deinitialized" our window. Any memory reserved for the window is made available for other uses and the window vanishes. Press CLEAR again and you should remain at the original screen.

Now let's make a 40 X 24 text window taking up the full screen. Type in the following lines:

```
iniz w1 <ENTER>
wcreate /w1 -s=01 00 00 40
24 02 08 08 <ENTER>
shell i=/w1& <ENTER>
```

Press CLEAR and you'll find yourself in a full screen, 40 X 24 text window, with an OS9: prompt. Remember, whenever you define a new device window and start a shell in it, OS9 treats that window as if it were an entirely separate computer. Whatever you do in one window has no effect on any other window.

You are probably wondering how we defined this new window with the WCREATE statement. It may look complicated, but the WCREATE statement is actually easy to use once you understand what all those numbers stand for. The syntax for WCREATE is as follows:

```
wcreate /dname -s=windowtype
xstart ystart colwidth numrows
foreground background border
```

/dname is the device name, in this case /w1.

-s=windowtype tells OS9 what type of window this is going to be. There are eight different types of windows, ranging from 01, which is a 40 X 24 text window which we used here, to 08 which is a high resolution graphics window.

Xstart and Ystart tell OS9 the location of the upper left corner of the window. Xstart is the column number and ystart is the row number. Since we wanted a full sized screen, we used 00 00.

Cwidth is the column width of the window. We wanted a full sized window so we used 40. Numrow is the number of rows the window will cover. We used 24 for a full screen.

Continued next page

OS9. You could start a program running here, press CLEAR to jump back into /w1 and run another program there, while the first program is still running on the original screen.

/w1 is pretty small, but there is no need to be restricted to this size window. OS9 lets us make our window any size we wish. All we have to do is tell it what we want.

OS9 has default specifications for windows w1 - w7. If, as in the case above, we do not define the type, size and colors we want to use, OS9 selects the default specifications for the window. But we can easily change this.

First, let's get rid of /w1. Press CLEAR until you are in /w1, then type:

```
ex <ENTER>
```

Your keyboard just went dead, didn't it? What you just did was kill the shell that had been operating in /w1. Without the shell you can't type anything any longer. Now press CLEAR and you should jump back to the original screen. Type this line:

uses a series of hexadecimal numbers for parameters that define the window we want to use.

OS9 has many window and graphics functions that are accessed through the DISPLAY function. The usual syntax is:

```
display    code1    code2
parameterlist
```

To create an overlay window, we use an OS9 command known as OWSet, Overlay Window Set. This would be typed as:

```
display 1b 22 parameter list
```

The 1b 22 is the OWSet command. This tells OS9 that we want to set up an overlay window. The parameters are similar to those used with WCREATE except we do not need to tell OS9 what type of window to use. The overlay window inherits the same window type as the device window it is overlaying. The parameters are:

```
SVS CPX CPY SZX SZY PRN1 PRN2
```

SVS is the "save switch". This tells OS9 if the material overlayed by the new window is to be saved or not. If you use 00, the material on the original screen covered by the new window will not be saved, and when the overlay window is removed, it will not be restored. If you set it to 01, the material will be saved and automatically restored when the overlay is removed.

CPX and CPY are the upper left corner of the overlay window, in hex. If you want the window to start in the far upper left corner of the screen, you would set CPX and CPY to 00 00. SZX and SZY are the size of the window in columns and rows (again in hex). If you want a window 10 characters wide by three rows or lines long, you would enter 0A 03.

PRN1 and PRN2 are the background and foreground palette registers to use for the foreground and background colors for the window. Note that these are palette register numbers, not color numbers. Let's see how it works. Type this line:

```
display 1b 22 01 07 09 0a 0a 09
0c <ENTER>
```

Neat, isn't it? All of a sudden you have a square window overlaying what was originally on the screen. Let's look at each of the parameters and see what they did.

1b 22 told OS9 we wanted to create an overlay window. The 01 told it we wanted to save what was under the window and restore it when we end the window later. The 07 and 09 tell OS9 the column and row where the box should start. In this case, column 7, row 9. The 0A 0A gives a box that is 10 characters wide by 10 rows deep (10 in hex is 0A). Finally, we selected color 09 for the background and 0C for the foreground.

To remove an overlay window, just type this at the OS9 prompt:

```
display 1b 23 <ENTER>
```

No parameters are needed. If you have more than one overlay window active, only the most recent window is removed. What would you use this for? The uses are limited if you're just sitting at the keyboard as we are now, but from within a program the possibilities are fantastic. You could use overlay windows as pop-up help windows, displaying information for the user quickly and easily while preserving the original screen display. It could be used for menus of functions or sub-menus, or even for data entry, because the window "inherits" the shell from the device window they are overlaying.

This article has by no means explored all of the possibilities of OS9's windowing system. But hopefully this will have proved that using windows under OS9 Level Two is rather easy.

If you have any questions or comments, or have suggestions about topics you'd like to see discussed here, please write to me at the address given below. If I feel a question might be of interest to other readers, I'll deal with it in the column.

We've covered a lot of territory and skimmed over some of it pretty fast. I hope I didn't hopelessly confuse you. Next time we'll take things at a slower pace.

The last three items are the colors you want the window to use. Foregnd is the color for the text, backgrnd is the background color and border is the border color. A table of color numbers is given on page 7-12 of the Getting Started section of the OS9 documentation.

Your OS9 system disk has several windows pre-defined for you in what are known as procedure files. They will create windows for you. These are: window.t38s, window.t80s and window.glr4. To use them, just type the name of the file at the OS9 prompt.

Window.t38s will build a 38 X 24 window. Window.t80s gives an 80 X 24 window, but it's hard to read on a composite monitor or tv unless you change the colors.

Window.glr4 is interesting. This procedure file puts three device windows on your screen at the same time. Note that /w3, the small window in the lower right corner of the screen, does not have a shell active in it. You can start a shell in this window by typing shell i=/w3&. Then all three windows will be "live."

Now let's look at the second type of window OS9 features, the overlay window.

An overlay window does just that, it overlays an existing device window or another overlay window, but the initial overlay window must be on a device window. Overlay windows are more "traditional" types of windows. They are used primarily for displaying information such as prompts, tables, etc. But the overlay window also "inherits" the shell from the device window it is overlaying, so it can also be used to enter commands and run programs.

One important feature of OS9's overlay windows is that they can automatically restore the text or graphics they disrupted when they popped up. When you remove an overlay window when you're done with it, you don't have to worry about re-building.

Again, the best way to show how it works is with a demonstration.

First, you can't use overlay windows on the 32 column display you see when OS9 first boots up on the Coco 3. (Unless you've created a new boot disk with CONFIG, OS9 boots up in the 32 column "VDG" mode that emulates the Coco 2.) You can only display overlay windows on top of a device window. So if you haven't done so already, make a 40 x 24 window described above and then use it for these examples.

We're going to be using the DISPLAY command to create and modify overlay windows. Display

CoCo Clipboard Coupons

Pascal Programming

Delmar Searles

from temporary storage to the vacant slot:

```
3 4 12 1 5 2
```

The first two numbers are sorted. Put the third number in temporary storage:

```
      12
3 4   1 5 2
```

Again there are no larger numbers to the left of the vacant position so no shifting is done:

```
3 4 12 1 5 2
```

The first three numbers are sorted. Put the fourth number in temporary storage:

```
      1
3 4 12   5 2
```

The third number is bigger, so move it to the right:

```
      1
3 4   12 5 2
```

The second number is bigger than the one in temporary storage, so move it to the right also:

```
      1
3   4 12 5 2
```

The first number is also bigger, so move it:

```
      1
3 4 12 5 2
```

Finally, move the number in temporary storage to the vacant position:

```
1 3 4 12 5 2
```

The first four numbers are sorted. Put the fifth number in temporary storage:

```
      5
1 3 4 12   2
```

The fourth number is bigger so move it right:

```
      5
1 3 4   12 2
```

Since the 5 is bigger than the 4, place the 5 in the vacant position:

```
1 3 4 5 12 2
```

The first five numbers are sorted. Put the last number in temporary storage:

```
      2
1 3 4 5 12
```

Shift the 12 down:

```
      2
1 3 4 5   12
```

Shift the 5 down:

```
      2
1 3 4   5 12
```

Shift the 4 down:

```
      2
1 3   4 5 12
```

Shift the 3 down:

```
      2
1   3 4 5 12
```

Finally, place the 2 into the vacant slot:

```
1 2 3 4 5 12
```

The entire list has been sorted.

Notice that as each number was placed in temporary storage, larger numbers in the already sorted portion were shifted to the right and the number in temporary storage was inserted into its proper location.

One of the most useful tasks performed by computers is sorting. There are a large number of sorting techniques available and efforts are being made to improve existing methods and find new ones.

To most of us, the time required to sort a file is not extremely important. Files are small enough that even inefficient sorting routines are adequate. In a business environment where files may easily contain tens of thousands of records the amount of time required to sort a file can be substantial.

Simple, easy to understand, sorting techniques tend to be rather slow. Fast sorting routines are difficult to explain (especially in writing) and hard to understand. We will take a look at two simple sorting techniques and write the corresponding Pascal code for each.

Insertion Sort

Many of you are already familiar with an insertion sort but may not know it. Card players often use a similar technique to put their cards in order. I'll use a row of numbers to represent the data being sorted. When a number has been placed in a temporary storage location, I'll write it above the rest of the line. Suppose this is the unsorted list:

```
3 4 12 1 5 2
```

Put the second number in temporary storage:

```
      4
3   12 1 5 2
```

Starting at the vacant position and working from right to left, shift any larger numbers one position to the right. In this case, there are no larger numbers. Move the number

Continued next page

Selection Sort

The selection sort works on a different principle. The largest element in the list is found by simply searching through the list from beginning to end. The result is marked by an asterisk (*):

```
3 4 12 1 5 2
      *
```

The largest element is then swapped with the last element in the list:

```
3 4 2 1 5 12
```

This guarantees that the largest element is last in the list. It is ignored from now on.

The largest element in the remainder of the list is found and swapped with the element in the next to the last position:

```
3 4 2 1 5 12
      *
```

Notice that nothing changes since it was already in its correct position. Now the last two elements are sorted. These two elements are ignored from here on.

The largest element in the remaining list is found

```
3 4 2 1 5 12
      *
```

and swapped with the element third from the end:

```
3 1 2 4 5 12
```

The last three elements are sorted and ignored from now on. The largest element in the remaining list is swapped with the element fourth from the end:

```
3 1 2 4 5 12
      *
```

```
2 1 3 4 5 12
```

The last four numbers are sorted. Finally, the largest number in the remaining list is swapped with the element fifth from the end:

```
2 1 3 4 5 12
      *
```

```
1 2 3 4 5 12
```

And the list is entirely sorted.

Some Comparisons

Neither of these two sorts is very fast. In both cases the typical execution time is proportional to the square of the number of elements in the list. This means that doubling the size of a list will quadruple the amount of time required to sort it.

The insertion sort is a good choice if you need to sort a list that is almost sorted already. Suppose you are working with a

sorted file and you update it by appending some new records to the end of the file. The insertion sort would be a pretty good choice for sorting the updated file.

The selection sort's primary advantage is the relatively small number of data shifts that must be made. It is a good choice when sorting files that contain large records. Moving these records around in memory can be quite time consuming. By reducing the number of record transfers in memory, the overall sorting time can be reduced.

Pascal Implementation

Listing 1 is a short Pascal program that illustrates the use of both of these sorting techniques. I first created a file of integers using a short BASIC program:

```
10 OPEN "0", #1,
"NUMBERS/DAT"
20 FOR I = 1 TO 100
30 PRINT #1, RND(100)
40 NEXT I
50 CLOSE #1
```

I used BASIC because of the convenience of the RND function, and because I could do it quickly.

The Pascal program reads this number file into an array of integers. The unsorted array is sent to the printer. The numbers are then sorted using the insertion sort and the selection sort. Each sorting routine prints out the sorted list.

```
program SortsDemo(input, output);
type
  dataArray = array [1..100] of integer;
var
  NumberList: dataArray; (* Array of data
elements *)
  ListLength: integer; (* Length of data file
*)
  (*****
  *
  * Read the data file
  *
  *
  *****)
procedure ReadFile(var NumberList: dataArray;
var ListLength: integer);
const
  InfileName = 'NUMBERS/DAT'; (* Physical data
file *)
var
  Infile: text; (* Logical name of data file *)
begin
  reset(Infile, InfileName);
  ListLength := 0;
  while not eof(Infile) do begin
    ListLength := ListLength + 1;
    readln(Infile, NumberList[ListLength])
  end; (* while *)
  close(Infile)
end;
```

I should point out that in this program the array of integers is passed as a value parameter. That is, the sorting routines work with a local copy of the original list. I did this to ensure that each sorting routine was given the same unsorted list of values.

Passing an array as a value parameter wastes a lot of memory since there will be two copies of the array in memory at the same time. When the insertion sort is called, a second copy of the array is created, sorted, and printed. At the end of the insertion sort procedure the memory used by this second array is returned to the available memory pool.

But then the selection sort procedure is called. Again a copy of the list is created, sorted, and printed. When the selection sort procedure is finished the additional memory is once more returned to the available pool.

In a practical program you will more likely want to pass the array as a "var" type parameter. That way the sorting routine will work on the original list rather than on a duplicate. It is not very often that you want to retain both the original list and the sorted list at the same time.

The sorting routines in the demonstration program use the entire data element when comparisons are made. In some cases you will want to sort a list of records based on the contents of a single field called the key field. We'll explore that topic in the next issue of CoCo Clipboard.

S I
u n
b f
s o
c
r P
i a
p g
t e
i
o 1
n 5

```

(* ***** *)
* * * * *
* * Print the data in the file
* * * * *
* * ***** *)
procedure Print(var NumberList: DataArray;
               Length: integer);

var
  LineCount, (* Number of Data elements printed
*)
    (* * in current line of output *)
  (* ***** *)
* * Selection Sort Routine
* * * * *
* * ***** *)
procedure SelectionSort(List: DataArray;
                       Length: integer);

var
  Max, (* Largest element *)
  IndexOfMax, (* Position of largest element
*)
  I, J: integer; (* Indicate position within
list *)
begin
  for I := Length downto 2 do begin
    Max := List[I];
    IndexOfMax := I;
    for J := 2 to I do
      if List[J] > Max then begin
        Max := List[J];
        IndexOfMax := J;
      end; (* if *)
    List[IndexOfMax] := List[I];
    List[I] := Max;
    end; (* for *)
    Print(List, Length)
  end;
  (*
-----
  M A I N D R I V E R
-----
*)
begin
  ReadFile(NumberList, ListLength);
  Print(NumberList, ListLength);
  InsertionSort(NumberList, ListLength);
  SelectionSort(NumberList, ListLength)
end.

```

```

I: integer; (* Loop counter *)
Outfile: text; (* Logical name of output file
*)

```

```

begin
  rewrite(Outfile, '-2');
  I := 0;
  LineCount := 0;
  while I < Length do begin
    I := I + 1;
    write(Outfile, NumberList[I]:4);
    LineCount := LineCount + 1;
    if LineCount = 20 then begin
      writeln(Outfile);
      LineCount := 0;
    end (* if *)
  end; (* while *)
  writeln(Outfile);
  writeln(Outfile);
  close(Outfile)
end;

```

```

(* ***** *)
* * Insertion Sort Routine
* * * * *
* * ***** *)

```

```

procedure InsertionSort(List: DataArray;
                       Length: integer);

```

```

var
  Temp, (* Temporary storage for data
element *)
  I, J: integer; (* Indicate position within
list *)

```

```

begin
  for I := 1 to Length-1 do begin
    J := I;
    Temp := List[I+1];
    while (Temp <= List[J]) AND (J >= 1) do begin
      List[J+1] := List[J];
      J := J - 1;
    end; (* while *)
    List[J+1] := Temp;
    end; (* for *)
  Print(List, Length)
end;

```

The Assembly Line

Kraig Brockschmidt

Welcome to the Assembly Line. Here I will present short assembly language programs of interest and application. All programs will be given in a BASIC program as well as a fully commented EDTASM+ source code file. I would appreciate your response, and perhaps could answer some of your questions. Please send all correspondence to:

Kraig Brockschmidt
14024 152nd Ave. S.E.
Renton, WA 98056-7313

For a personal reply, please include an SASE. If I get enough questions about a certain topic I will present here--on The Assembly Line.

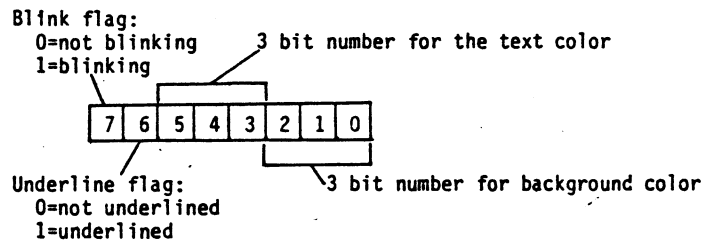
Some programs I have seen for the Tandy 1000 line (like Tandy's Deskmate) put a frame, or 'window' around a block of text on a hi-res text screen. The structure of the new CoCo III hi-res text screens make it easy for the CoCo to make windows.

The program in Listing 1A is a short example of how to work with the new text screens. It fills the 80-column screen with garbage characters and puts a 9*5 character box around some text. A joystick moves the box around the screen. Listing 1A is fully commented, so I will only point out the special features of the text screens.

The main difference between the 32 and 40/80-column screens (besides the width) is the way the characters are stored. Each character uses only one byte on the 32-column screen but uses two bytes on the others. The first byte on a 40/80-column screen contains the code for the character, and the next byte is its attribute. Figure 1 shows what each bit in the attribute byte controls.

To produce the window, several ROM routines and values stored in high memory are used. The most important is the value stored in

FIGURE 1: The Attribute Byte



EXAMPLES

10110010=178 ATTR 2,6,8
00001111=15 ATTR 7,1
11010110=214 ATTR 3,2,U,8

\$FE08 the current attribute byte. Any character sent to the screen via JSR [\$A002] (or when a CLS is done) the characters (or screen) will have this attribute. The program initially sets \$FE08 to 26, the equivalent of ATTR 3,2. If you store characters and attributes directly to the screen you don't have to worry about \$FE08. The other high memory value used is at \$FE04, which contains the width of the current screen. If you write a program that switches screens you will need to monitor this value.

There are several ROM routines you will want to know about:

JSR \$F652 Set 32-columns
and CLS.
JSR \$F65C Set 40-columns
and CLS with the attribute in
\$FE08.
JSR \$F679 Set 80-columns
and CLS with the attribute in
\$FE08.

These are the exact routines used when you enter a WIDTH command from BASIC. In the program the screen is set to 80-columns with an attribute byte of 26, the same as ATTR 3,2. Our window will have an attribute of 60, or ATTR 7,4.

The biggest problem using these text screens is that they are located outside the 6809's 64K address space. Using the memory management unit (MMU) in the GIME chip we can effectively 'move' the text screen into the 64K area, store characters to it, and 'move' it back. In all reality, nothing moves. I will explain this thoroughly in a later issue. Nevertheless, the CoCo III's new ROM has two routines to control access to the screens. A JSR \$F772 opens access and JSR \$F778 closes access.

When you open access, the text screens are addressed through the block of memory \$2000-\$3FFF. The first location on the screen is at \$2000. Note that whatever was in that memory range is unaffected; however, DO NOT run a machine language program from this area because it will immediately try to execute the text screen! You also cannot access what was originally in this area!

Continued next page

The subroutine in Listing 1A called JUNK (line 830) shows a quick use of these routines. First, we open access to the screen. We fill it with data from the Super Extended BASIC ROM, loaded from \$E000 on and stored at \$2000 on. When the screen is full, we close access, returning the 64k address space to normal. Another note: The 80 column screen, even though we move it into an 8k RAM area, only uses 3.75k of memory (look at line 880).

Making the window only requires that we open access to the text screen, find out where to start the window, and change the attributes of each character in the window. A subroutine called BOX (line 590) draws a 9*5 character box using the attribute in the A register. It is used both to set and reset a box, since when we move the box we remove the one already on the screen (lines 500-550).

The final thing I want to point out is the four instructions at lines 340-370. First we store the B register on the S stack. We add that value on S to B again, which doubles B. Next we shift the value on S to the right, dividing

it by two, and add it to B again. Overall, this multiplies B by 2.5. Remember that only integer math is allowed in assembly language. More instructions will allow multiplication by any number. At first, B is read from the horizontal joystick and limited to be 0-57. Multiplying by 2.5 fits it to the horizontal width of the screen - 160 memory locations.

The rest of the program is explained in Listig 1A, and I wish you fun using the great CoCo III text screens in your assembly language programs.

```

10 *****
   * HI-RES TEXT WINDOW *
   * COPYRIGHT (C)1987 *
   *BY KRAIG BROCKSCHMIDT*
   *14024 152ND AVE. S.E.*
   *RENTON, WA 98056-7313*
   *****
20 CLS: CLEAR200, &H7000:
   FORT=1TO159: READ A$:
   X=X+VAL("&H"+A$): NEXT: IF
   X<>16386 THEN PRINT "ERROR IN
   DATA ": END
30 RESTORE: FORT=1TO159: READ A$
   :POKE &H6FFF+T,
   VAL("&H"+A$):NEXT :PRINT "DATA
   IN MEMORY" : PRINT "START =
   &H7000, END =&H70AB, EXEC=&H7000"
40 END
50 DATA 86, 1A, B7, FE, 08, BD,
   F6, 79, 17, 00, 7E, 9E, 8A, 34,
   10, AD, 9F, A0, 0A, B6, 01, 5B,
   44, 81, 13

```

```

60 DATA25, 02, 86, 13, F6, FE,
   04, 58, 3D, 1F, 01, F6, 01, 5A,
   C1, 39, 23, 02, C6, 39, 34, 04,
   EB, E4, 64
70 DATA E4, EB, E0, C4, FE, 3A,
   30, 89, 20, 00, AC, E4, 27, CF,
   1F, 12, 35, 10, B6, FE, 08, 8D,
   0A, 86, 3C
80 DATA 1F, 21, 34, 10, 8D, 02,
   20, BC, BD, F7, 72, CE, 00, 09,
   C6, 01, A7, 85, 34, 10, 30, 89,
   02, 80, A7
90 DATA 85, 35, 10, 5C, 5C, 33,
   5F, 11, 83, 00, 00, 26, EA, CE,
   00, 04, A7, 01, A7, 88, 11, 30,
   89, 00, A0
100 DATA 33, 5F, 11, 83, 00, 00,
   26, EF, BD, F7, 78, 39, BD, F7,
   72, 8E, 20, 00, CE, E0, 00, A6,
   C0, A7, 81
110 DATA 8C, 2F, 00, 25, F7, BD,
   F7, 78, 39

```

Listings next page

SPECIAL DEAL ON 500 PROGRAMS!

GET 50 DISKS OR 50 CASSETTE TAPES FULL OF OVER 500 PROGRAMS. HERE IS WHAT YOU'LL RECEIVE:

- * Over 250 Utility/Home Application Programs including a Word Processor, DataBase, Spreadsheet, Account Manager, 2 Basic Compilers, Terminal Programs, ROM Copies, Mail List, Machine Language Tutorials, Plus Much More!
- * Over 200 exciting games including Warlords, Star Trek, Super Vaders, Solar Conquest, Horse Races, Football, Baseball, Frog Jump, Invader, Plus Much More! (Many machine language games)
- * Over 30 adventures including The College Adventure, Dungeon Master, Space Lab, Ice World, Ship Wreck, Zigma Experiment. Plus 32K Graphic Adventures.

EACH INDIVIDUAL ISSUE SOLD FOR \$9.00 EACH OR \$450 FOR ALL 50 ISSUES. WE SLASHED THE PRICE TO ONLY 150.00.

REG. \$450 **NOW** \$150.00

★★ THIS MONTH ONLY ★★

Buy this package of 500 programs and receive a free 6 month subscription. (A *35 value)



Available on COCO 1, 2 and 3!
All Programs Include Documentation!



THE GREATEST SOFTWARE DEAL ON EARTH JUST GOT BETTER!

THAT'S RIGHT! THIS MONTH WE'VE DROPPED OUR YEARLY SUBSCRIPTION RATE AN UNBELIEVABLE *10.00 TO ENTICE YOU INTO SUBSCRIBING WITH US. GET 12 DISKS OR TAPES A YEAR CONTAINING OVER 120 QUALITY PROGRAMS. A SUBSCRIPTION TO T & D SOFTWARE CONSISTS OF 10 READY-TO-LOAD PROGRAMS DELIVERED BY FIRST CLASS MAIL EVERY MONTH. NO WE ARE NOT THE SAME AS THE RAINBOW ON TAPE. IN FACT, MANY SUBSCRIBERS HAVE WRITTEN IN AND SAID THAT WE ARE MUCH BETTER THAN RAINBOW ON TAPE!



	TAPE	THIS MONTH ONLY
1 YEAR (12 Issues)	78.00	60.00
6 MO. (6 Issues)	48.00	35.00
1 ISSUE	8.00	8.00

* Michigan Residents Add 4%
Overseas Add \$10 to Subscription Price
Personal Checks Welcome!

- * 16K-64K Color Computer
- * Over 4000 Satisfied Customers
- * Back Issues Available From
- * July '82 (Over 500 Programs)

- OUR LATEST ISSUE CONTAINED
1. Accounts Receivable
 2. Work Mate
 3. Calendar
 4. Invasion
 5. Trip Adventure
 6. Foot Race
 7. Flippy the Seal
 8. Screen Calculator
 9. Able Builders
 10. Super Error 2

```

00470 TFRX,Y*STORE NEW X
00480 PULSX*GET OLD ONE
00490 LDA$FE08*GET BACKGROUND ATTRIBUTE
00500 BSRBOX*RESET THE BOX
00510 LDA#60*BYTE FOR ATTR 7,4
00520 TFRY,X*GET NEW X
00530 PSHSX*SAVE IT FOR NEW COMPARISONS
00540 BSRBOX*SET THE BOX
00550 BRALOOP*REPEAT MAIN LOOP
00560 *MAKE A 9*5 BOX OF ATTRIBUTE BYTE A WITH THE
UPPER LEFT
00570 *CORNER POINTED TO BY X.
00580 BOXJSR$F772*SET UP HI-RES TEXT
00590 LDU#9*DO 9 COLUMNS
00600 LDB#1*POINTER
00610 HORISTAB,X*STORE ATTRIBUTE B BYTES FROM X
00620 PSHSX*SAVE X
00630 LEAX640,X*SHIFT IT DOWN 4 LINES
00640 STAB,X*STORE ATTRIBUTE AGAIN
00650 PULSX*GET X BACK
00660 INCB*ADD 2 TO
00670 INCB*POINTER
00680 LEAU-1,U*DECREMENT COUNTER
00690 CMPU#0*IS IT DONE?
00700 BNEHORI*CONTINUE IF NOT
00710 LDU#4*FOUR REPETITIONS
00720 VERTSTA1,X*STORE ATTRIBUTE IN FIRST COLUMN
00730 STA17,X*STORE IT IN 9 COLUMNS OVER
00740 LEAX160,X*MOVE X DOWN A LINE
00750 LEAU-1,U*DECREMENT COUNTER
00760 CMPU#0*ONE YET?
00770 BNEVERT*CONTINUE IF NOT
00780 JSR$F778*COME OUT OF HI-RES TEXT MODE
00790 RTS*RETURN TO MAIN LOOP
00800 *FILL SCREEN WITH CHARACTERS FROM
SUPER-EXTENDED BASIC ROM
00810 JUNKJSR$F772*SET UP HI-RES TEXT.
00820 LDX#$2000*GET START OF SCREEN
00830 LDU#$E000*POINT TO JUNK DATA
00840 LDA,U+*GET A JUNK BYTE
00850 STA,X++*STORE ONLY THE CHARACTER
00860 CMPX#$2F00*DONE WITH SCREEN? 80-COL. SCREEN
IS 3.75K
00870 BLOJUNK+9 *CONTINUE IF NOT
00880 JSR$F778*COME OUT OF HI-RES TEXT MODE
00890 RTS*RETURN TO LOOP
00900 ENDXX*END OF PROGRAM

```

```

00010 *****
00020 * HI-RES TEXT WINDOW *
00030 *BY KRAIG BROCKSCHMIDT*
00040 *14024 152ND AVE. S.E.*
00050 *RENTON, WA 98056-7313*
00060 * FOR THE 'CLIPBOARD' *
00070 *****
00100 ORG$7000
00110 *SET UP 80 COLUMN SCREEN, ATTR3,2
00120 XXLDA#26*GET NUMBER FOR ATTR 3,2
00130 STA$FE08*STORE IT
00140 JSR$F679*SET 80-COLUMN SCREEN
00150 LBSRJUNK*FILL THE SCREEN WITH JUNK
00160 LDX$8A*X=0
00170 PSHSX*SAVE FOR COMPARISON
00180 *MAIN LOOP TO FIND JOYSTICK SCREEN POSITION
00190 LOOPJSR[$A00A]*READ JOYSTICKS
00200 LDA$15*GET VERTICAL JOYSTICK
00210 LSRA*MAKE IT 0-31
00220 CMPA#19*IS THE VERTICAL TOO BIG?
00230 BLOVOK*BRANCH IF NOT
00240 LDA#19*MAKE IT SMALLER
00250 VOKLDB$FE04*# OF COLUMNS
00260 LSLB*DOUBLE IT. HI-RES SCREEN
00270 * TAKES TWO BYTES PER CHARACTER.
00280 MUL*D=BYTES DOWN
00290 TFRD,X*PUT D INTO X
00300 LDB$15A*GET HORIZ JOYSTICK
00310 CMPB#57*IS IT TOO BIG?
00320 BLSHOK*BRANCH IF NOT
00330 LDB#57*MAKE IT SMALLER
00340 HOKPSHSB*SAVE B
00350 ADDB,S*MUL B BY 2
00360 LSR,S*MAKE CONTENTS ON S HALF
00370 ADDB,S+*ADD TO B TO MAKE B=B*2.5; CLEAN UP
STACK TOO
00380 ANDB#$FE*CLEAR BIT 0---ASSURES AN EVEN NUMBER
00390 ABX*ADD INTO X
00400 LEAX$2000,X*ADD START OF HI-RES TEXT SCREEN
00410 *NOTE THAT SCREEN STARTS AT $2000 AFTER THE
JSR $F772 COMMAND.
00420 *BE SURE OF THE MMU REGISTERS YOU SET!
00430 CMPX,S*HAS THE JOYSTICK MOVED?
00440 BEQLOOP*READ JOYSTICKS AGAIN IF NOT.
00450 *BY CHECKING FOR JOYSTICK MOVEMENT, RAPID
BLINKING
00460 *OF THE BOX IS AVOIDED.

```

On Line with CompuServe

Wayne Day

"There's an international Color Computer club in existence, today, that boasts over 5,000 members. The club has no dues, meets almost 24 hours a day, 7 days a week, and has the largest collection of public domain software for the Color Computer anywhere."

It's funny how those words, originally written almost four years ago, and for another new Color Computer magazine, probably are as true today as they were then.

The "club" has a new name -- it's now known as "The CoCo Forum" instead of "*The Color SIG*", but with the exception of us growing larger each year, it's still pretty much the same friendly place it was back then.

The club still meets 24 hours a day on the CompuServe Information Service, the largest consumer oriented electronic information service in the world.

My name is Wayne Day, and I'm the managing SYSOP (Systems Operator) of the CoCo Forum, along with the other TandyNet forums on CompuServe.

Through these bi-monthly columns, I'd like to be able to introduce you to telecommunicating in general, and hope to take you on a fairly extensive tour of the CoCo Forum if you're not a CompuServe subscriber, and to help you use the service more effectively if you already are one of our members.

CompuServe is a commercial information system of computers, headquartered in Columbus, Ohio. Through the CompuServe Information Service, it provides electronic news and information on a subscription basis, to anyone with access to a telephone and a terminal, or a personal computer with the appropriate software and a modem.

Although most users of CIS are in the United States and Canada, the CoCo Forum and the OS-9 Forum both boast members on all continents of the world, including some very active newcomers from CompuServe's sister service "Nifty-Serve", in Japan.

CompuServe maintains its own communications network with dial-up access numbers throughout the United States, and additionally uses the facilities of the Telenet and TymNet services to supplement our own network. Canadian subscribers call in via the Datapac service of Bell Canada. Overseas, Telenet and CSCNet (Computer Sciences Corporation) access extend the availability of CIS.

Thus, there are very few points in the world where you won't be able to access CompuServe.

What's there, once you get on?

A wide variety of services including online banking and shopping... Wall Street quotations and financial research... the latest news from the Associated Press and National Weather Service weather bulletins radar maps, and of course, the discussion forums.

The CoCo Forum is one of those discussion forums where we provide a common meeting ground for folks who have the same interests as you, and have an interest in meeting other CoCo owners.

The message sections look very much like a local bulletin board would, only a lot busier -- usually, you can ask for help in a question left on the message section, and you'll have a reply within minutes. Sometimes, though, a message will require some research, and your question on the CoCo Forum will be seen by folks who are experts in their particular interest... that gives you a much broader exposure to your message.

The forum also provides a real-time interactive place where you can "chat" with other users called Conference.

The Conference hall can handle hundreds of conversations at the same time, but you'll usually find one or two groups of folks huddled together discussing something of mutual interest. Sometimes, though, it's a late-night gab session where we can all get together with our other friends, and "let our hair down" and get silly. Irregardless, the ability to "chat" with more than one person, electronically, is something that a local BBS usually can't offer since they are inherently single-user systems.

The data libraries are an instant hit with most folks -- here's where you'll find the reference files, programs, music and art work of many hundreds of folks who want to share their work with you. The listing of the available files filled over 500 pages the last time we printed it all out for a convention -- thousands of individual files, arranged in several logical sections so you can limit your search to only those subjects you're interested in. We'll tell you more about that next month.

Think you might be interested in joining us online? Well, there's several ways to get a CompuServe membership.

Radio Shack sells two packages.. one with software, and one without. Ask for the Universal Startup Kit if you already have a terminal program for your computer.

Bookstores and computer shops will also be seen selling the CompuServe Starter Kit -- it comes with a printed guide to the service

GOLD

A Subscription to CoCo Clipboard Magazine

is worth more than gold. All meat articles - no fluff, no filler. Wayne Day on CompuServe! Kraig Brockschmidt, Randy Krippner, Bob van der Poel and Delmar Searles on programming! Jim DeStafeno and Rush Caley on Business and Mike Dooley on Ham Radio! Coming soon - Bill Bernico and BASIC HELP and CoCo Club Corner!

And don't forget **CLIPBOARD COUPONS!** Discounts, free disks, free catalogs and more. You'll save enough to pay for your Clipboard subscription with the coupons in this issue alone!

A subscription is just **\$12.00** for 6 issues. Your check or money order is welcome! Canadian subscriptions are also **\$12.00** when paid in U.S. funds. C'mon - join the adventure!

Yes! I'm ready to join the adventure. Enclosed is my check or money order for **\$12.00** for 6 issues (1 year) of CoCo Clipboard Magazine.

Name _____

Address _____

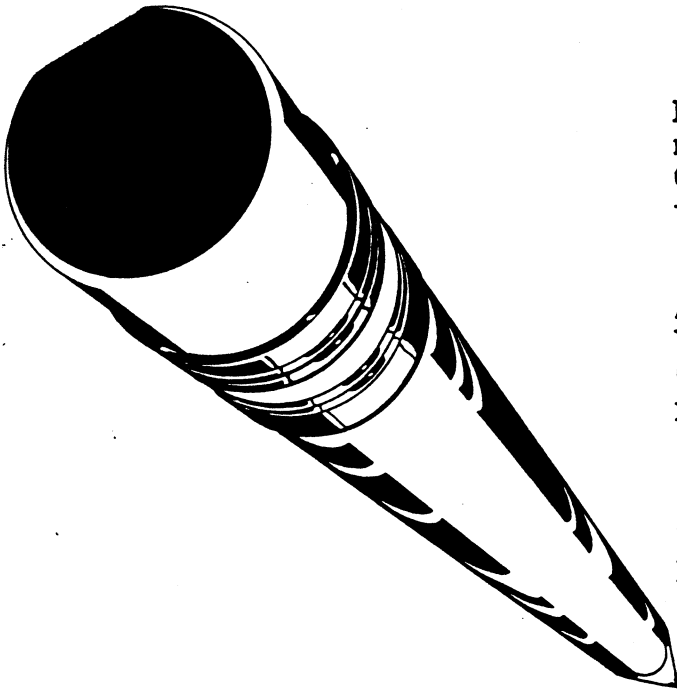
City _____

State or Province _____

ZIP or Postal Code _____

Mail to:

Note: It's not necessary to use this form to subscribe if you need to use the coupons on the next page. A photocopy is fine or just a note in the mail along with your check.



MARTY'S RUBBER STAMP DISCOUNT COUPON!

Colored Printer Ribbons for:

Okidata 80, 82, 83, 92, 93, 8300, CP-100, CP210
Microline 80, 81, 82, 82A, 83, 83A, 92 & 93.

Star Delta 10, 15, 15F, DP8480. Gemini 10, 10x,
15, 15x, 15xf, SG10, 10c, 15.

6 @ \$3.00 ea. / 12 @ \$2.50 ea.
(add \$2.50 shipping)

Marty's Rubber Stamp Shop
100 Pike
Port Carbon, PA 17965
(717) 622-7077

SIGMA SOFTWARE COUPON

THIS COUPON IS GOOD FOR:

5% OFF ON ANY 2 PROGRAMS or

10% OFF ON ALL THREE PROGRAMS

in our ad in this issue of CoCo Clipboard
Magazine.

Send your order with this coupon to:

SIGMA SOFTWARE
14024 152ND. AVE. SE
RENTON, WASHINGTON 98056

Save now on Emerson Computer Services Software.
Order any program and if your order says, "I SAW
IT IN COCO CLIPBOARD" you take advantage of
the following discounts:

1. \$0.00 to \$49.99 take 5% off total order!
2. \$50.00 to \$99.99 take 10% off total order!
3. OVER \$100.00 take 15% off total order!

On a \$100.00 order you'll save enough to pay for
your subscription to CoCo Clipboard. Discounts do
not apply to shipping charges and must be added
after your discount.

T&D Subscription
Software,
an advertiser
in this magazine,
wants to introduce you
to their variety of
inexpensive software
by giving you a:

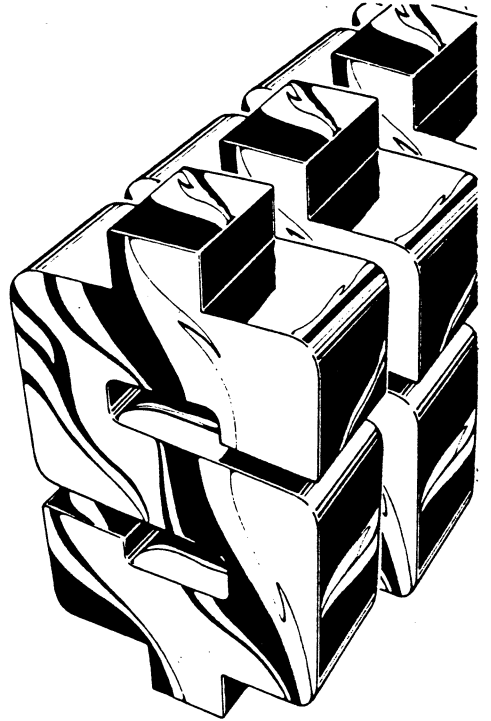
FREE Disk or Tape
Filled with 10 programs

1 coupon per household

Circle one: TAPE DISK

Mail this coupon to:

T&D Free Offer
P.O. Box 1256
Holland, MI 49423



Three Dollar DISCOUNT COUPON
Include with your order for a \$3.00 discount
Or mail to Tothian Software, Inc. Box 663
Rimersburg, PA 16248, for our 1988 Catalog
and a three dollar discount coupon.



Name-----
Address-----

CoCo 'N Amateur Radio

Mike Dooley

Hello, and welcome to our new magazine! This column is going to look at the Radio Shack Color Computer (here-in-after called COCO) and how it is used in Amateur Radio.

Amateur Radio is a service in which ordinary people communicate across town and around the world. It started back in the early days of radio and has been growing ever since. Today's Amateur Radio operators (called Hams) can communicate with each other using any one of several modes of operation. The two earliest methods of communication were voice and Morse code. As time went on, and the hobby grew, other modes of communication were adopted. Radioteletype (RTTY), facsimile (FAX) and even television (ATV) have been developed. Even satellite communications are used!

Each new mode gave Hams something to experiment with and learn about. Computers, when they became available, were added to that list of things to experiment with in the Ham Shack. The longer they stayed, the more useful they became.

In the early days of Radioteletype older mechanical printers were used. They were big, heavy and required mechanical adjustments and oil. Computers could perform the same functions of the mechanical beasts (except give us a printed copy) and were much neater and cleaner! Soon, newer and better methods of sending RTTY messages were found!

First it was faster speeds, then new modes of operation such as AMTOR and Packet. These two modes can send data error free over the airwaves! Morse code could be transmitted using a computer to key the transmitter. Some smart Ham even discovered a way to receive Morse code and print it on the screen using the same computer!

Facsimile is now done with computers. Facsimile is a high resolution still picture. One of the more common items distributed using FAX are weather charts. News photographs, also, are sometimes sent via FAX. These pictures were originally sent using a mechanical device with a rotating drum. The computer has replaced the old FAX machine in this process.

Computers however have not made the same dent in ATV (Amateur Television) as they have in Radioteletype or FAX. They're mostly used to produce pictures and graphics for transmission from one Ham to another much the same as a camera and pen and paper were used.

In all this discussion of the things Amateur Radio operators can do with computers one question has become obvious. Do you need to be a licensed Amateur Radio Operator to enjoy these many things? Well, yes and no. If you only want to monitor, no license is needed. It's only when you want to transmit over the airwaves, and there are only certain bands of airwaves Amateurs are allowed to transmit on, that you need the license. There are many things that can be monitored using a computer and most General Coverage Receivers. RTTY transmissions sent via the various news services from all over the world is just one example.

If you're interested in becoming a Ham there are several ways to learn how. There's the Amateur Radio Relay League. They're located at:

225 Main St.
Newington, Connecticut 06111.

Their phone number is 203-666-1541. They can put you in touch with the Amateur Radio club or the Volunteer Examiner group in your area. Another good way to find out about becoming a Ham is to find another Ham. If you don't know one already (and we're everywhere) then,

while you're out driving around town, watch the other cars around you. See the one with five or more antennas (especially the strange looking antennas)? Follow him, because you've probably just found a ham. If he can't help you become a Ham personally, he'll know who can.

It has suddenly become very obvious... computers can do almost anything around the Ham Shack. They can even keep the Logbook. All that is required is an enterprising person to write the software and/or build the hardware needed to perform most of these functions. What we're going to do in this column is discuss and enjoy any software and/or hardware available for the COCO that can be used for Morse code, RTTY, FAX or anything else interesting around the old Ham Shack. That's what this column is all about.

We'll concentrate on software and hardware available to get you on the air in the easiest manner. We'll look at software available through stores and the public domain. Hardware can be either purchased or we may build it ourselves.

In the next installment we'll look at using the COCO to get on the air with Packet Radio. Until then... 73's!

(Editors note: If you would like to contact Mike about using your CoCo with Amateur Radio send your letters to CoCo Clipboard.)

```

PROCEDURE hdir
0000
0001 (* this module opens the directory as a
read device and
0038 (* passes the path name and the name of
the root directory
0072 (* the the recursive routine which does
all the work
00A6
00A7 TYPE registers=cc,a,b,dp:BYTE;
x,y,u: INTEGER
00CC TYPE diskspecs=cdate(5):BYTE;
dname:STRING[32]
00E7 DIM regs:registers
00F0 DIM diskid:diskspecs
00F9 DIM path:BYTE
0100 DIM indentlevel,t: INTEGER
010B DIM firstlevel:STRING[50]
0117 DIM temp:STRING[50]
0123 DIM root:BOOLEAN
012A indentlevel=0
012B
0132
0133 (* before doing anything we have to
find out
0160 (* which device this routine is being
called from.
0192 (* This could be 'W1' or 'TERM', etc.
The I$GetStt
01C4 (* call is used with SS.DevNm. This
returns the name
01F8 (* of the current device for the
specified path. In this
0230 (* case this is standard input. We open
that device for
0267 (* output so that the title can be
displayed.
0294 (* By writing the program in this
manner the packed
02C8 (* procedure can redirect the output
when called.
02F9
02FA regs.a=0
0305 regs.b=$0E
0311 regs.x=ADDR(temp)
031F RUN syscall($8D,regs)
032D RUN hdir2(temp)
0337
0338 OPEN #path,"/"+temp
0346 PRINT #path,CHR$(12)
0350 FOR t=1 TO 9
0360 READ temp
0365 PRINT #path,temp
036F NEXT t
037A
037B (* get some user input
0391
0392 INPUT #path,">",firstlevel
03A0 CLOSE #path
03A6 IF firstlevel="" THEN
END
03B2 ENDIF
03B4 RUN hdir3(firstlevel)
03B6
03C0
03C1
03E1
03E2
03EA temp=firstlevel
03F6 OPEN #path,temp:DIR
regs.a=path
0402 regs.b=$0E
040E regs.x=ADDR(temp)
041C RUN syscall($8D,regs)
042A RUN hdir2(temp)
0434 CLOSE #path
043A OPEN #path,"/"+temp+"@"
044C IF firstlevel="/"+temp THEN
045D root=TRUE
0463 ELSE
0467 root=FALSE
046D ENDIF
046F
0470 (* print the disk ID
0484
0485
0495 FOR t=1 TO 80
049B PRINT "-";
04A6 NEXT t
04A7 IF root=TRUE THEN
04B2 SEEK #path,$1A

```

```

04BC      GET #path,diskid
04C6      RUN hdir2(diskid.dname)
04D3      PRINT "Directory of '"; diskid.dname;
"; Created: ";
04FB      PRINT diskid.cdate(1); "/";
diskid.cdate(2); "/"; diskid.cdate (3)
051F      ELSE
0523      PRINT "Directory of: "; firstlevel
0539      ENDIF
053B      PRINT
053D      (* Everything's all set up, go read the
053E      directory
056F      RUN hdir1(firstlevel,path,indentlevel)
0570      CLOSE #path
0584      PRINT
058A      END
058C
058E
058F      DATA "Hierarchical Disk
Directory",",", "(c) 1987 by Bob van der Poel"
05D3      DATA "17435 - 57 Avenue", "Edmonton,
Alberta", "Canada T6M 1E1"
0610      DATA "", "Input directory to process
(eg. '/d0', '/h1', ",', '/d0/cmds',
etc.)."

PROCEDURE hdir1
0000
0001      (* This routine expects the following
to be setup:
0033      (* 1. The disk must be set up for
raw-read (rootpath)
0068      (* 2. A complete pathlist to the
directory to be read (direntry)
00A8      (* 3. The current indentation level
must be set (indentlevel)
00E5      PARAM direntry:STRING[32];
00E6      rootpath:BYTE; indentlevel:INTEGER
00FE      (* The structure 'idinfo' matches the
00FF      data in the file ID sector.
0141      (* The program could access additional
information and report more
0183      (* stats if desired.
0197
0198      TYPE idinfo=att:BYTE; own:INTEGER;
modate(5):BYTE; link:BYTE
; filesize(4):BYTE; cdate(3):BYTE
DIM filetype:INTEGER
DIM fileinfo:idinfo
01D0
01D7
01E0
01E1      DIM entry:STRING[32]
01ED      DIM isdir,blankline:BOOLEAN
01F8      DIM path:BYTE
01FF      DIM idsect:REAL
0206      isdir=FALSE
020C      (* Open the current directory so we can
020D      read all the filenames
024C      (* Skip the first two entries (. and
..)
0274
0275      FOR filetype=1 TO 2
0285      OPEN #path,direntry:READ+DIR
0291      SEEK #path,64
029A      (* handle each entry in the directory
029B      WHILE NOT(EOF(#path)) DO
02C0
02C1      GET #path,entry
02CC      IF LEFT$(entry,1)<>CHR$(0) THEN
02D6      (* get logical sector number of
02E6      ID sector for this file
02E7      idsect=ASC(MID$(entry,30,1))*655
031E      36.+ASC(MID$(entry,31
031F      ,1))*$0100+ASC(MID$(entry,32,32)
)
034F      RUN hdir2(entry)
0350
035A
035B      (* get the ID information ---
needed to see if this is a DIR
0396
0397      SEEK #rootpath,idsect*256
03A6      GET #rootpath,fileinfo

```

Is A Hard Drive In Your Future?

Jim DeStafeno

What comes in a dark buff steel jacket and is faster than a speeding electron? Give up? Well... it's a Hard Disk Drive. The one I have is by Owl-Ware. Take a tip up front, if you are a single floppy man thinking about a second drive or a have twin floppy system, thinking of a ram disk or a "tape only" man thinking of going to a floppy, read on. You may be able to skip one and gain two.

I must admit that I have never used a Ram Disk on a CoCo, nor am I a OS-9 user, but I have done some reading, "To get greater speed out of your disk drive, go to 64K and ML it to a 6ms track to track speed." "Be sure that you get the XXX software for your Ram Disk so that it will...". "If you are planning to do any serious programming in OS-9 you should really have a twin disk (floppy) drive and a 40 or even an 80 tracker." My two cents worth is to skip it all and go to THE solution, a hard disk. What's on the flip side? Check out the following and you can be the judge. Remember now, here I am referring to just the Owl-Ware offering. Upon ordering, I had a few questions about compatibility with BASIC as I don't do much with anything else. I was told that all would go well in BASIC using the CoCo-II or "I", but it was not ready to run on the "III". That was okay by me since I am not ready to run the III either. They added that if I were to run OS-9, their HD would work on any of the CoCo's.

They said delivery would require a couple of weeks due to the slow production of the interface. The only strange thing was their request that I not use a credit card. Something was said about fraudulent use and the FBI. After a bit of hesitation I said okay. After hanging up I dashed off a note and enclosed a check. Several weeks later I was told that their 30MB HD was not ready for BASIC so I OK'ed a down grade to 20MB. A couple of weeks later a

check arrived for the difference in cost; but no HD.

After another couple of weeks I called to find out where the HD was. I was told a mix-up had occurred; the down grade was miscommunicated internally as a cancellation. About a week later my UPS man came to the door with the goodie box.

I popped the box open and dug out the instructions. There are two sets. One is written by the interface supplier, LR Tech and the other is by Owl-Ware. The LR Tech sheets seem to address the OS-9 user. Lots of advise, addresses, etc. The other set begins talking about the BASIC user right away. Somehow, somewhere I got the idea that I had to install the interface inside of the HD case. The interface does NOT go inside of the HD case, nor is one expected to take the cover off the case. However I did find two things of interest inside.

The first one concerns the entry of a two inch wide ribbon cable into the back of the HD case. The cable goes through a slot made to mount a connector in and it has no grommet. Bells and whistles went off in the back of my head! Reckoning back to my old military aircraft days; "Don't ever, ever, ever let a wire chaff on the edge of metal. It will wear away the insulation and cause a short." No shorts for my HD; I wrapped the cable where it passes through the slot with tape. I even thought about buying a male and female connector, but after further investigation I saw that the cable runs from the back to the very front of the case, about fifteen inches. Since I have the HD along side the monitor, the cable runs from the computer along the outside of the case, around the rear, into the case through the slot and then all the way to the front of the case. I expect I will cut a slot

in the plastic bezel on the front and run the cable straight out to the CoCo. However that will wait until the system gets several hard hours on it.

I call it all a system because all HD's have three major components. There is the Hard Disk itself. It is only five or so inches in diameter. The case also contains a PC board two thirds the size of the CoCo board. This board is called the "HD controller". It has its own power supply and computer to keep the i/o's straight and tell the mechanical parts what to do.

Since all computer request codes are not the same, an interface is needed to convert the computer request codes to the HD's codes. The ribbon cable that goes in the back of the case comes from the interface. It's another PC board in a metal box. It looks very much like a floppy disk controller. Both it and the floppy controller are plugged into a "Y" cable.

There is one more trick needed. The computer has to be told by a program to tell the HD what to do. RS disk BASIC has no commands to talk with more than four drives, which are reserved for the FD drives. Owl-Ware has modified the possible drive numbers up to nine and elected to use the Winchester HD instructions. In turn they have been over-layed into RS disk BASIC. Viola, a language we all know adapted to operate a HD. I have had zero errors since I pressed the first key. Good job Owl-Ware!

The SAVEing organization is very good. As with the floppies, a SAVE must be given a name and directory number; there is no automatic default to the HD, unless you define it with DRIVEN. There are 6 directories; DIR-4 through DIR-9; but none of them are limited to a

size. That is, if a 20MB program could be SAVEed to the HD, it would take it. Along the same logic, one could save all programs to one directory until the HD is full. Then why 6 directories? To get faster action.

The action first goes to the requested directory and then the directory list is searched until the requested program, or file, is found. If there were 120 programs in one directory, and the requested program is the last one in the directory, a search of 119 program names would be made before the request could be granted. However, if the programs were equally divided between the 6 directories, the longest search possible would be 20. So one can tailor for speed by saving given programs in given places in given directories. In addition, one could separate types of programs to help find them quickly. Utility programs in one directory, business programs in another, games in yet another and so on.

Owl-Ware says their HD system works through OS-9 code. You do NOT need OS-9 to use their Winchester BASIC. However, the saved file name can't have a space in it, I use a "-". Also, if lowercase letters are used in a program name, the letters screen print in real lowercase. I often use the RS inversed lowercase letters to advantage since they stand out so well, but am not able to here. Another difference is in the way the information is given upon a DIRn command. DIRn produces a double column display; twice as many listings on one screen AND it stops scrolling when the screen is full. A tap of the space bar yields another screen full; v-e-r-y nice.

Continuing with the needed goodies; Winchester BASIC comes in a ROM chip. Owl-Ware says the use of a multi-ROM chip floppy disk controller is a good idea. I'm not sure I agree, but see "thorns" below. I've had the Disto Super Controller since it first came out. It really has been "super" and it too has worked without flaw since day one.

The last item needed is a Multi-Pac Interface or a "Y" cable. I have a RS box and have hated it from day one. I opted to get a "Y" cable. Howard Medical advertised the lowest price and said it was for CoCo Max. I asked them if it was good for any "Y" cable need. They said yes. I sent for it and it came long before the HD. I noted that the contacts were silver color, not gold. Thought they had come up with something new. No way, after 8 hours of use I was getting I/Os. I mean the CoCo community has known for several years that solder contacts don't cut the mustard with the CoCo.

Anyway, as soon as the Howard Medical "Y" cable went off I ordered another from Spectrum Projects. I have been buying from Spectrum Projects for many years and have had zero problems. As an example, in this case I guess he was out of stock of "double Y"s because he sent me a triple for the price of a double. I am embarrassed to say that I went somewhere else first, but I think it is best to get the word out about this good service.

So what about the thorns on this rose bush? Well, there are only two, but they are sharp. One is that it uses part of normally free RAM for a buffer. It reduces available RAM by a little more than 1K. Even worse, the addresses that are used, are the same ones used by many ML programs. Guess I will get very good at off-set loading.

I wrote Owl-Ware suggesting several things, one of them being that they set-up a clearing house log of off-sets for popular ML programs. I Never did get a response from them. Anyway, if any of you Owl-Ware owners fine a useful off-set, send it in. I will list it here.

The second thorn has to with changes in some of the BASIC commands. Owl-Ware says BACKUP doesn't work. I havn't been able to get DSKI\$ and DSKO\$ to work either; and that is unfortunate. DSKI\$ allows the reading of program and file names from a given disk directory, into a program. All BASIC disk utility programs use that command, which means none of them work. More over, the ML disk utilities won't work because the format of a HD is different than a FD, floppy disk, and/or they won't allow a drive number greater than 3. That means that there is no way to mass copy the programs in a HD directory to a FD. Of course LOAD and SAVE, and KILL and RENAME still work so anything can be done; it just has to be done one program at a time.

I guess DSKINI doesn't work. The HD comes formatted and there is no need to do it again. However, a program is included to do it; I hope I never have to use it. By the way, Owl-Ware sells a couple of utility programs, which I bought. I was a bit unhappy when I didn't find a floppy in the shipping container. I laughed at myself when I found everything in directory 4 of the HD. One of the utilities, written in BASIC, mass copies the contents of any FD directory to a HD directory. You see, the commands that don't work when a "from" HD drive is specified, still work when a "from" FD drive is specified, even if the "go to" drive is a HD drive.

As I said, I don't like the Multi-Pac Interface for several reason; space consumed, it's looks and its action. In my opinion, the "Y" cable is only slightly better. I mean, it and the connected stuff flop around like a half dead fish. Anyway I was pleased to see Bob's "Y" cable. It is long enough to run it out of the trap door port, up and over the top of the case. A neat 90 degree fold upon itself allows the HD interface to be attached to the upper right hand corner of the case with velcro, with the ribbon cable exiting to the rear. Then the FD controller can lay on top of it, also with its cable exiting to the rear.

T-64 won't work with such a set-up, not even if I unplug the HD interface. That means unplugging the FD controller from the "Y" and the "Y" from the CoCo and shoving the controller in the trap door. Good night, there has to be a better way. Well, not all is lost. You see, I was talking to the Disto controller people about the fact that their 80-Column screen display doesn't work with Winchester BASIC. The 80-C is activated through their C-DOS. As with Winchester BASIC, C-DOS is overlayed on RS-DOS. There seems to be no conflict

Continued next page

"CoCoPack"
from Bill Bernico Software

PROGRAMS! PROGRAMS! PROGRAMS!
63 programs for only \$6.00.
That's about 9 1/2 cents per
program. Included are music,
graphics, utilities, disk,
educational, home help, printer,
games, even a word processor and
21 font styles written in BASIC
that you can use in your own
BASIC programs. Send check or
money order only to Bill Bernico
Software 708 Michigan Ave.
Sheboygan, WI 53081

now scans the new directory and handles all the entries in the same manner as before. Again, if a directory entry is found HDIR1 is called again. This process continues until all the sub-directories have been processed. Each time a sub-directory is finished, the program ends and control is returned to the calling point. If this is the last entry in the root directory control will return to HDIR; if it was recursive call the "CHD '..'" will reset the current directory to the parent of "dentry." This permits the rest of the current directory to be scanned.

The remaining modules are subroutines used by HDIR and HDIR1. The first, HDIR2, converts data passed to it to a string. The second, HDIR2, converts a string of upper and lower case letter to all uppercase.

When typing in the programs do not type in the numbers in the lefthand column. These are the hex addresses where the code resides in

memory and are used mainly in debugging. Also, do not indent the lines -- this formatting is handled by the BASIC09 list utility.

The programs presented here contain lots of comments, code segments are set off with blank lines and variable names have been kept as "meaningful" as possible. If you do the same with your BASIC09 programs you'll be doing yourself a favor -- not only will your code be easier to debug and maintain, but you'll probably end up understanding the tasks better and thereby write a better program. Oh, and if you're wondering about how to insert blank lines, just type 2 spaces after the "E:" prompt -- the first space is the insert line command, the second is the line's contents (a single space).

The programs can be tested from the command mode in BASIC09 with the command "RUN HDIR." However, this will only display the directories on the screen. If you wish to use all the features of the program you should pack the modules into executable code. From BASIC09 command mode, with all 4 modules in memory type:

```
PACK      HDIR, HDIR1, HDIR2, HDIR3
>HDIR
```

This will compress the programs and place them in a file called HDIR in your current execution directory. Now you can call the program from the OS9 prompt (or from within a program) with the simple command "HDIR." If you want the output of the program to go to the printer, use "HDIR >/p." When calling HDIR make sure that both SYSCALL and RUNB are present in your current execution directory.

Since a program is never finished, I'll finish this article off with a challenge to you. Have a look at the variable "fileinfo" in HDIR1. This variable contains not only the file attribute information, but also details of the file owner, the creation date of the file, the filesize, etc. Can you include this additional data in your printout? No five dollar bills for the winner of this one -- just a feeling of satisfaction for those who participate. Remember, the only way to learn to program is TO PROGRAM.

DeStafeno cont.

between them, but they are in different ROMs, therefore can not be normally loaded together. However Disto has two solutions.

The first is a f-r-e-e driver program, sent on a floppy. It is designed to lay the 80-C calls over any DOS. There is also a driver for the clock and parallel port. It arrived in two days, but I couldn't get it to LOAD. Probably my fault, but I copied Winchester BASIC to their floppy and sent it all back to them with a cry for help. My letter to Owl-Ware asked for help on the 80-C problem too. If they have a solution, they are keeping it to themselves.

The second solution is much better. In fact, it solves two problems. In a couple of months they intend to release their own BASIC interface that will fit right inside the Disto FD controller. (They already have one for OS-9.) The solution is to scrap the one Owl-Ware sent and install the Disto. Expected price, \$50. It will allow the full C-DOS compliment to run with the HD. I didn't ask about BACKUP and DSKI\$/DSKO\$. However, I wouldn't be surprised if they worked just fine.

So... the question stands, should everyone get a HD? Most CoCo users smile at our power-to-dollar ratio. A HD is costly, but under many circumstances I think it is false

economy to get something else. First I guess I had better explain my roots. I cut my computer teeth on a punch card chewing "main frame". Then I went on to work with the best big blue had to offer, and now sell automated business systems driven by an excellent super micro, and of course have the CoCo micro at home.

I really feel that true business applications can not be done properly without a HD. Sounds a bit like a snob? Remember when you were on a tape only system and the big spenders said the only way to go was by disk? My argument follows along the same lines. A proper business system is composed of many little programs. Most of the systems I sell have four and five hundred programs, plus thirty and forty files. Such a situation on a floppy system would have one playing CoCo Frisbee until the cows come home.

In addition, I'm sure that any true business application with the CoCo will (or should) be done under OS-9, with "C" or BASIC 09. For now I'll limp around in BASIC, but on a HD. There are ways to make BASIC work fairly fast, but a lot of specific program planning has to be done to get it all to happen right. On top of a HD being mandatory, an 80 column display is mandatory too.

Continued next page

HGRX-DUMP \$13.95

The 1ST CoCo III screen dump that prints ALL PMODE/HSCREENs to Tandy & Epson printers. Menu baud rates double-size/reverse, and 16 print patterns. Print just PMODEs on a CoCo I or II! 100% ML, disk/tape.

Sigmadisk \$14.95

Disk utility for all CoCos. Three directories, copy, backup, format, directory backup & editor, sector editor, gran table, 40/80 col. on CoCo III, more! 100% ML on disk. Supports 1-4 drives. A must!

Sigmaword \$9.95

A great low-priced word-processor for all CoCos. 50*22 screen, full-screen edit, disk I/O, 15K buffer w/ large file capability, margins, justification. 80% ML, disk only.



Ask about games!
No shipping. Send
check/money ord.
PAYABLE TO:

Kraig Brockschmidt
SIGMA SOFTWARE
14024 152nd Ave. SE
Renton, WA 98056

Another \$100 for 80 column yields a total of \$430.

However, I bet Van der Poel would fix that in a wink of an eye.)

Most everyone will agree the next hardware investment after a disk is a monitor. If one is running a monitor, I think it is silly to run anything less than an 80 column display. With the normal 32 x 16 display there are 512 character spaces to communicate information to the user. An 80 X 24 display has 1920; 3.75 times more information space. The super micro I sell has 132 X 24, 6 times more.

If he were to get a HD he would need the Disto FD controller at \$100 with C-DOS, a Disto HD interface at \$50 and the Disto 80 column at \$100. That totals to \$250, leaving \$180 left over from the twin FD cost. I expect a 10MB HD and HD controller can be gotten for about \$350 and no dead fish or watermelon wart will be needed, just the normal wart. (I expect that means Telewriter-64 will work except for the drive number.

So it seems that in a couple of months a 10MB HD that will support an 80 column display on a "-2" can be gotten for \$600; \$170 more than a twin FD. That's many hundreds fold increase in memory space at a 28% increase in cost. Very difficult for the serious user to turn down.

However, any purchased program that is to be used on a Tape/HD only system would have to be loaded in

continues page 26

Does all this mean a CoCo-3? If one doesn't have his CoCo yet, of course. Forget the low price on the "-2". In two years all the best programs, utilities and games will be on OS-9 Level 2; windows and all that, you know. How about the fellow that already has a "-2"? I'm at the machine hours on end with no unusual eye fatigue. An investment of about \$135 and a half hour of wire clip-on work inside the CoCo, no soldering, gets it all. Also I havn't found any Hi-Speed or Super Hi-Speed Poke errors with my "-2". So for home use, a "-2" is good enough until the second generation OS-9 Level 2 business programs come out. The only lost investment would be \$100 for the Disto 80 column, but you would be able to carry it over to the "-3", using the clock and parallel port that comes with the Disto 80 column display. Neither of them come with the "-3" or Level 2, and OS-9 always wants to know what time it is. As you can see not all of the \$100 investment is "lost".

Of course if you already have a FD system without a Disto you would have to shell out another \$100 for it, but I include that in the price of the HD, and of course the old FD controller could be sold for \$45 or more. And if the system is a twin, one of the drives could be sold for sure. If it's a single drive, it may be best to hold on to it for convenience. The fellow thinking about a RAM disk is directly comparable to the fellow with a FD system. He can do the same and have the hundred plus dollars cost of the RAM disk available to invest in the hard. What would he be losing? Not much just a little longer time between the end of one program and the beginning of the next. RAM disks do load their contents into working RAM very quickly, but a HD ain't no slow poke.

Now what about the fellow that is presently on a tape only system? Well lets take a look at it. He is going to need a FD controller, cable, drive and manual. The best price I see is \$180; but that is for a single drive. I don't know any serious user that is happy with a single drive. The best priced twin I see is \$330, and that is with a Disto controller too.

COLORED RIBBONS
for the DMP-130 OWNER



NOBODY ELSE HAS THEM

1 3 6

black 10.50 8.00 7.00
COLOR 13.50 13.00 12.00

GREEN, BROWN, RED, PURPLE, BLUE
SHIPPING AND HANDLING SEND \$2.00

We have very, very
LOW PRICES on paper,
ribbons, disks, and
other supplies. SEND
\$1 for our FULL-COLOR
catalog and we will
refund a \$2 credit!



MARTY'S RUBBER STAMP SHOP
47 SOUTH SECOND ST. P.O. BOX 165
SAINT CLAIR, PA 17970
(717) 429-1052

PROCEDURE hdir3

0000

0001

(* this subroutine will convert a string to all uppercase letters

0042

0043

004F

005E

0065

0066

006F

0079

007A

008E

009B

00AE

00B9

00C6

00C8

PARAM text:STRING[500]

DIM t,where,length:INTEGER

DIM char:BYTE

length=LEN(text)

where=ADDR(text)

FOR t=0 TO length-1

char=PEEK(where+t)

IF char>96 AND char<123 THEN

char=char-32

POKE where+t,char

ENDIF

NEXT t

```

03B0 (* If this is a directory AND
03B1 we've displayed all the
03E7 (* files in the directory then
print the directory name
041E (* and do a dir. of it. This is a
recursive call.
044F
0450 IF fileinfo.att>127 THEN
045F isdir=TRUE \(* set flag =
second pass req'd
0484 IF filetype=2 THEN
0490
0491 IF POS>indentlevel THEN
049C PRINT
049E ENDIF
04A0 PRINT
04A2 FOR t=1 TO indentlevel
04B5 PRINT " ";
04BB NEXT t
04C6 RUN hdir3(entry)
04D0 PRINT entry; " ****"
04DD indentlevel=indentlevel+3
04E8 CHD direntry
04ED RUN
hdir1(entry, rootpath, indentlevel)
0501 CHD "...
0507 indentlevel=indentlevel-3
0512 ENDIF
0514 PRINT " ";
0516 ENDIF
0517 IF fileinfo.att<128 AND
filetype=1 THEN
052D
052E (* If not a DIR then it must be
a filename -- print it.
0565
0566 IF POS+LEN(entry)>78 THEN
0575 PRINT
0577 ENDIF
0579 IF POS<indentlevel THEN
0584 FOR t=POS TO indentlevel
0596 PRINT " ";
059C NEXT t
05A7 ENDIF
05A9 PRINT entry.
05AF ENDIF
05B1
05B2 ENDIF
05B4 ENDWHILE
05B8 (* Close input directory. If this is
05B9 pass 1 then check to see if
05F9 (* any directories were found --- if
true then loop and handle them.
063C CLOSE #path
063D
0643 IF POS>1 THEN
0644 PRINT
064E ENDIF
0650
0652 EXITIF isdir=FALSE THEN
0653 ENDEXIT
065E NEXT filetype
0662 END
066D
PROCEDURE hdir2
0000
0001 (* This routine parses a string with
the last bit set to indicate
0042 (* its end. It clears bit 7 of the last
character in the string &
0083 (* adds a $FF to mark the end of the
string.
00AF
00B0 PARAM setstring:STRING[300]
00BC DIM where:INTEGER
00C3 where=ADDR(setstring)
00C4
00CE WHILE PEEK(where)<$80 DO
00CF where=where+1
00DD ENDWHILE
00E8
00EC POKE where, PEEK(where)-$80
00ED POKE where+1, $FF
00FB END
0107
0109

```


CoCo-izing Your Business

Rush T. Caley

Editors note: The first two parts of Rush's articles on CoCo-izing Your Office appeared in Spectrogram Magazine. If you would like reprints of the first two articles please contact CoCo Clipboard.

I.) USER INFLUENCES

Before discussing any guidelines in software selection, let me say that the choosing of software for any purpose, business or otherwise, is essentially a series of trade-offs. If money is a consideration, one might define a price bracket to which he must confine choices. If one is limited in time to invest, he might restrict himself to less sophisticated programs that do not require the hours steeped in documentation. But choices do not always limit in a negative sense; sometimes they increase possibilities rather than limit them.

Software selection can also be a matter of taste. In that case, it is a matter of a person's individuality or personality. Oftentimes it may even reflect strong and weak points in a person's conceptual or manual capacities. I have a tendency to notice the practicality of software and its raw power. I don't get too caught up in the graphics or niceties of the "bells and whistles" until I evaluate how well and how completely it does the job it was designed to do. If too much code is tied up in how the program looks, I know I'm trading space available for more working power. That tendency is a function of my personality.

I avoid spreadsheets like a winter flu. I am probably the worst spreadsheeter (is that a word?) on the planet. But when it comes to database management, that's where I shine. This is a perfect example of how our capabilities are reflected in our software choices. I cannot explain my lack of understanding of spreadsheets. Nor can I understand my expertise in database design. It is just the way my head works. But those facts will influence the manner in which I attack a problem, and what choices I might make in software.

The influences mentioned above are always in the background when software selection is at hand. But in a business environment, many more practical considerations come into play. As I related earlier in this series, it is extremely important that decisions have been made as to what data be managed and how you want it managed. The software selected should always keep the total plan in mind. Here is an example that is not at all far-fetched:

+++++

PROBLEM :

You need to mail letters to clients and/or prospects.

SOLUTION :

You decide to purchase a mailing list program.

QUESTION :

Are you merely going to print mailing labels, or do you want mail merge as well in order to write a specific letter to selected clients on the list?

NEW PROBLEM :

You need the capability to maintain a client address list, and you desire a mail merge feature so that during certain promotions, or for collections, you can select clients meeting certain criteria and mail them the same "boiler-plate" letter.

SOLUTION 2 :

You require a mail list program with a mail merge feature.

NEW QUESTION :

You have a client file with address information and a maillist program with mail merge. What other types of information other than addresses do you need to manage concerning those clients? Purchases? Monies owed? Next appointment? etc. Do you need a database program to handle this additional information.

NEW PROBLEM :

You need a program to manage assorted information for your business for all of your clients.

SOLUTION 3 :

You require a database program.

NEW QUESTION :

Do you buy a mail list program for labels and letters, and a database program for the other information? OR Do you buy a database program that has a text editor and mail merge feature built in? Finally, are you the only one who requires this information - or do you need the ability to share the information with others such as a branch office?

Continued next page

NEW PROBLEM :

You need to decide on the mail and database question. You also need the ability to share information stored in your files.

SOLUTION 4:

Decide the mail and database question. Also, purchase a TELECOMMUNICATIONS PACKAGE.

NEW PROBLEM : What features are needed

+++++

So you see, this can start out simply enough, but can mushroom into a seemingly endless series of decisions. This is the reason many people avoid computerization all together. They are overcome by choices and the apparent complexity of even the slightest of tasks. Anyone who tells you that moving from a manual system to computerized systems is simple, is grossly misinformed. But whether you are a novice or have experience with computers, the fundamentals are the same. Extrinsic factors such as taste, finances, and personal abilities will influence software choices to a degree. But intrinsic factors such as the way you choose to do business, the volume of the data, and the manner in which you wish to manage it, should be the major influence in your selection of software. And as you can see from the above example, the solutions you choose impact the abilities and effectiveness of your overall designs and intentions.

II.) SOFTWARE DESIGN INFLUENCES

Once we have come to a generic solution to our software needs, then comes the problem of choosing the specific software package that is designed for the function we have defined. If at all possible, it is best if we can have an opportunity to test a piece of software before we purchase it. However, this is not always possible. This is especially true in the Color Computer market where most software comes from third-party vendors. So it is not always quite as simple as running down to the "Shack" for a test drive. But obtaining as much information as possible prior to purchase is very important if we want to be sure that the package meets our requirements.

Every piece of software has its own built-in limitations and restrictions. These may be a result of the program designer, or of the CoCo itself. A good illustration of

this occurred when I was choosing accounting software for the place where I was business manager. I was desperately in need of an accounting package as manual record keeping was not only tedious, but restricting any possibility of proper reporting capabilities. I had seen an ad for an accounting package that looked as though it would fulfill all my requirements. It did double entry accounting, printed checks, had Income/Expense and Balance Sheet reports, and other goodies. It was relatively simple to use as it was menu driven. However, when it came down to meeting all of my needs, I was out of luck and out \$75 merely due to the intrinsic design of the program. Below are listed the problems that occurred.

1. The program allowed for 100 accounts, and I needed over 300!
2. The 100 accounts available were assigned to specific categories. 1-20 = assets; 21-40 = liabilities; 41-60 = Equity; 61-80 = Income; and 81-100 = Expenses. If I had more Expense accounts and did not need 20 Income accounts, I could not assign one of those numbers to an expense category because the program would not treat it as an expense.
3. The program allowed for a 3 digit account number, while my accounting system required 6.

This is just a brief sample of the problems you can run into if you do not know ahead of time exactly how you want your data managed. As I have said on many occasions, never allow a software program dictate how you will run your business. The software should be the servant and not the master. All of these influences of both the user and the program itself can frustrate one in the computerization of office procedures because of the many choices and trade-offs they impart. However, being aware of their existence and being equipped to meet the choices is two-thirds of the battle. Next month we will delve even deeper into the software.

DeStafeno cont.

via tape. That may just be an inconvenience rather than a problem. Once the program is on the HD it would run just as though it were on a FD. If I'm not in error, any program can be put on tape and loaded into the computer. From there it would be a simple matter to save it to the HD. Of course program and file back-ups would be put on tape. If that doesn't appeal, don't get more than one FD drive. When you get fed up playing CoCo Frisbee, spring for the HD.

If you are now calculating backward to see how a "-3" would fit into the matrix, don't forget that by the time you are done, OS-9 will cost near \$100 extra. A very good deal for sure, but as I said, if one is not using OS-9 I don't see any advantage of the III over the II, other than the 80 column display.

So I highly recommend a HD for the serious BASIC user, even before getting a III; surely over a second floppy disk drive. And if one were thinking about going to a RAM disk or going from a tape system to a disk system, I'd make it a Hard Disk. If you don't plan to stick with BASIC very long and are in a hurry, you couldn't go far wrong with the OWL-WARE system. However if BASIC is your thing, it would be a good move to wait and see what the Disto equipment from CRC Computer, Inc. offers. Either way, business needs mass storage and for my buck a hard disk is the very best way to get it.

W. Day cont.

as well as \$25 credit to get you started. Check your local B. Daltons or Waldenbooks, as well as Target, Kresge, K-Mart's and such. I've seen them on sale, locally, for less than \$20.

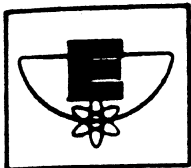
You'll need a CompuServe ID, a modem and some terminal software to get started -- next issue, we'll make sure you know how to put them all together!

(Editors note: To contact Wayne on CompuServe you can use "Easyplex" electronic mail. Wayne's I.D. is 76179-0074

In the Next Issue . . .

*Bill Bernico's
"Basic Help Column"*

*Ted Paul's
CoCo Club Corner*



EMERSON COMPUTER SERVICES CO.

8289 BANNER ROAD S.E. • PORT ORCHARD, WASHINGTON 98366 • (206) 857-7878

MEMO TO: ALL CLIPBOARD READERS
FROM: Rush T. Caley, OWNER

To those of you who know me and ECS, you know that we have been the Northwest's finest in Color Computer Support since 1983! We are the ONLY consulting firm devoted solely to the CoCo and its applications for BUSINESS and practical home uses. I appreciate your past patronage and look forward to serving you again with some of our exciting new offerings!

To those of you who do not know me or my company, I invite you to take a closer look. You can get many programs from any number of excellent vendors in the CoCo market. But ECS specializes in the programs you can't get ANYWHERE ELSE! That's because we promote the CoCo and we promote CoCo programmers that turn out quality CoCo application software. Ask around. Our service is next to none!

Take a look at these exclusive ECS offerings: The descriptions in our catalog are in much more detail. All on DISK only.

SCMS: SALES CLIENT MANAGEMENT SYSTEM **NEW 8/12/87**

The ultimate personal productivity tool for Sales Professionals. Client Listing, Daily Schedules, Neglected Client Report, and User-Defined reports that can search on up to 14 fields simultaneously! CC I, II, III \$49.95 + \$3.00 S&H.

PVDL: PERSONAL VIDED DATA LIBRARY ** NEW 7/10/87 **

Marvelous for keeping info on your video collection. Holds up to 200 titles per file. Tracks Title, Rating, Type, Director, Star and Co-Star, Year of release, run time, tape position, etc. CC I, II, III \$16.95 +\$2.00 S&H.

WORKBASE DATA SYSTEM PRODUCTS: * RAW POWER AT A TAME PRICE*

WORKBASE I & II relational database managers. All 9 menu - driven BUSINESS PORTFOLIOS: A/R, A/P, Payroll, Gen. Ledger, Sales Order and Purchase Order Entry, Inventory Control, Rental Property Mgt., Church Membership Mgt. ** CC I, II, III Database Managers \$79.95 ** Business Portfolios \$39.95 each * S&H \$3.00

CALINDEX II The Calendar/Scheduler/Project Manager. Up to 15 employees or individuals can keep track of projects with warnings of tasks past due. A great tool for any business, or organization like a church, club, or a family. A MUST for any project-based office. Discover patterns of efficiency. Monitor accountability. OR you can just keep track of your own appointments, things to do, etc. ** CC I, II, III \$42.00 and \$3.00 shipping ** REQUIRES 2 DISK DRIVES.

That's only a short look! Write or call for a catalog today! It will cost you \$2.00 for the catalog, BUT it applies to any purchase you might make. See coupon in this issue of CLIPBOARD for a special discount offer.

TOTHIAN HELPS THE TEACHER COPE!

GRADER 2 - Takes the work out of making out grades. You can average numerical grades with or without weighting, or you can average various components of a grade and calculate a weighted average of the components on-screen or printed. It gives a printout of the student's name or number, percentage grade, number of points, and total number of points. It also contains some analysis routines similar to, but not as powerful as, **DISTER 2**. 16K Tape or Disk - - - - - \$14.95

DISTER 2 - Does a statistical analysis of lists of numbers such as, but not limited to, grades. It works out averages, sample and population standard deviation, and variance. The data can be saved to tape or disk. 16K ECB Tape or Disk - - - - - \$14.95

SEATER 2 - With or without name lists from **ALPHER 2**, you can rapidly produce printed seating charts for classrooms of almost any size. The program creates a preliminary seating arrangement from the name list you give it. Then, with the game-like on-screen editing function, you can modify this preliminary chart to suit yourself. 16K ECB Tape or Disk - - - - - \$14.95

ALPHER 2 - Produces either printed or on-screen alphabetized lists of names (first name first or last name first) or words. These lists may be saved to tape or disk and later used with **GRADER 2** or **SEATER 2**. 16K ECB Tape or Disk - - - - - \$9.95

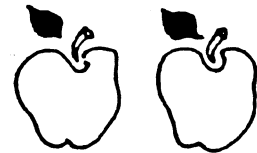
TESTEM II - 32K ECB program for creating multiple choice, completion, short answer, true or false, and matching tests. Randomizes questions if desired. Questions can be saved for future use. Prints answer key. Prints answer sheets for virtually any test format. Contains a heading function for including instructions on the test. 32K ECB Tape or Disk - - - - - \$19.95

TEACHER PAK PLUS - consists of **GRADER 2**, **DISTER 2**, **SEATER 2**, **ALPHER 2**, and **TESTEM II** on a single Tape or Disk. 32K ECB Tape or Disk - - - - - \$42.95

Add \$3.00 for UPS or COD. PA residents add 6% sales tax.



TOTHIAN SOFTWARE, INC.
Box 663
Rimersburg, PA 16248



CoCo
Clipboard Magazine

3742 U.S. 20 Box 3, • Fredonia, N.Y. 14063

BULK MAIL
U.S. POSTAGE
PAID
PERMIT NO. 18
FREDONIA, N.Y.
14063