For your TANDY Color Computer

Registered by Australian Post — Publication No. QBG 4009

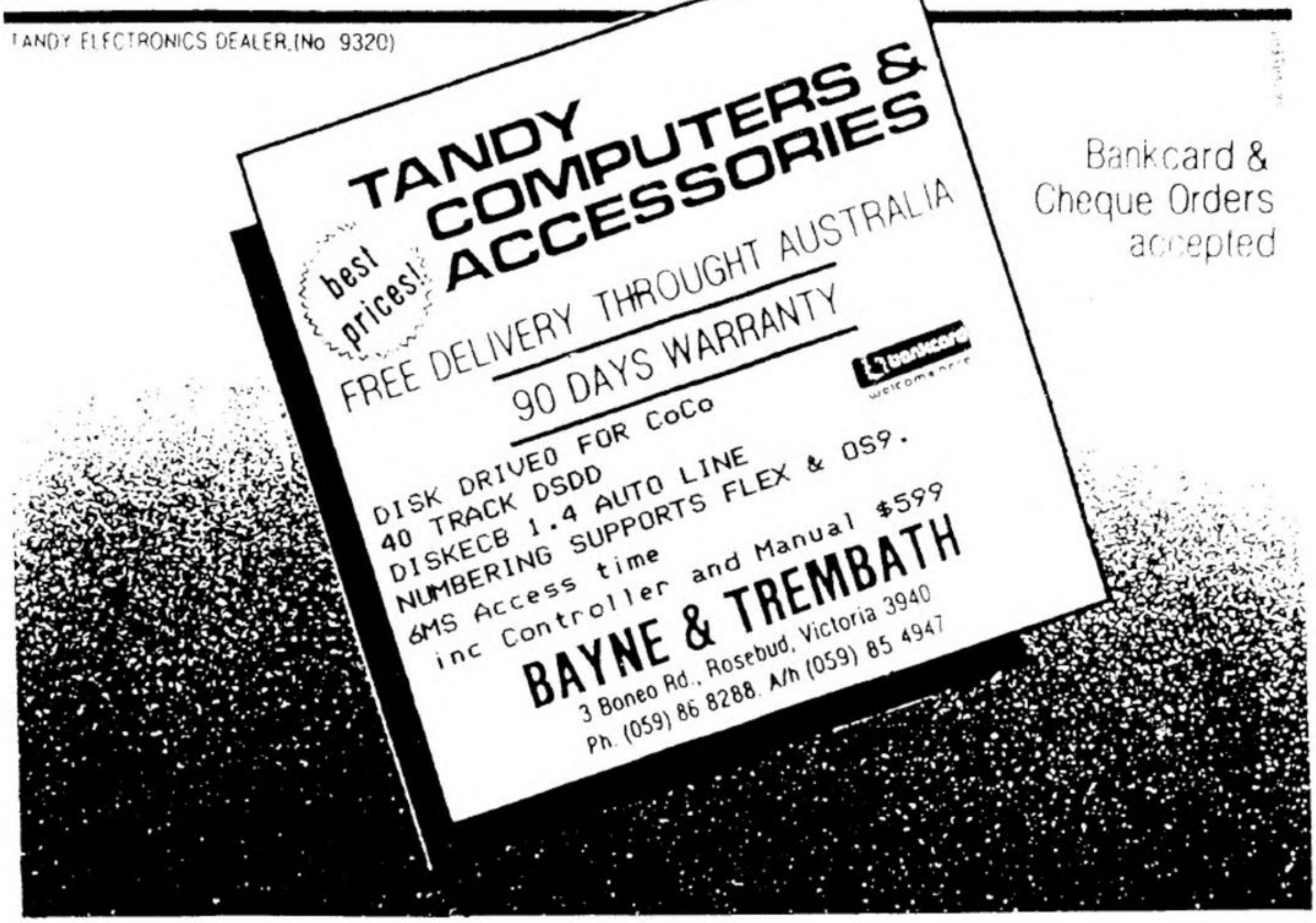
PNG K4.95 NZ $5.20

$4.50

AUSTRALIAN RAINBOW

January, 1986

No.55



AUSTRALIAN

RAINBOW

index

UPPERCASE = ARTICLE + PROGRAM
lowercase = article only

AUSTRALIAN RAINBOW Publisher and Editor Graham Morphett. Co- editor Kevin Mischewski. Assistant Editor Sonya Young. With grateful assistance from Brian Dougan, Richard and Judy, Bob Thomson, Paul Humphries, Alex Hartmann, Michael Horn, Jim and Sheryl Bentick, Annette Morphett. Cover Art Jim Bentick.

ADVERTISING DEADLINES: The 7th of the preceeding month of publication. All advertising is arranged through ToToAdvertising, PO Box 6730, Gold Coast Mail Centre, Qld. 4217

OS 9: Kevin Holmes is the contact for os 9 information. He also has access to OS 9 Software from the US. His address is: 39 Pearson St., Narara, NSW. 2250.

All programmes in this issue of AUSTRALIAN RAINBOW are available on Rainbow on Tape.

The contents of this magazine is COPYRIGHT. Magazine owners may maintain a copy of each programme plus two back ups, but may not provide others with copies of this magazine.

Telephone: 075 51 0015 Voice, 075 32 6370 CoCoLink

Printed by: Australian Rainbow Magazine PO Box 1747 Southport, Qld. 4216

Founder GREG WILSON

PRINT #-2,

We start this year with a new corporate identity.

As many will be aware, we took this magazine over under very sad circumstances, and everything we have done, has been coloured by the speed of that inital take over.

Since that time we have allowed the natural name "Australian Rainbow Magazine" to be the name by which we are known.

But we have become involved in an increasing number of endeavours which do not relate directly to Australian Rainbow Magazine.

The latest of these - 'softgold', our Tandy 1000 / IBM PC / Tandy 100 / Tandy 4 magazine which will be separating from Australian CoCo Magazine later in the year, is a prime example.

So we have adopted the corporate name "Goldsoft".

'Gold' because,

1. we deal with talent, a commodity more valued than gold;

and 2. our magazine is published on the beautiful Gold Coast of Australia.

And 'Soft', because

1. we don't like the plastic 'hard' feel of the large publishing houses' magazines;

and 2. we deal mostly in software for your computer!

The name change is nothing more than that. You are dealing with the same people who have the same responsiblilties and the same commitments to CoCo and the magazine.

*

Our world has been changed irrevocably by the computer.

Nowhere is this more apparent than in the area of communication, where even though you don't realise it, computers monitor and assist even a simple phone call to your neighbour.

When we began our association with this magazine back in July 1984, the Bulletin Board was a thing we all wondered about - wondered if there would be a use for one, wondered if people would want to use them.

Today they are a fact of life. And although only a presently small percentage of the CoCo community have modems, that percentage is growing, and this is the growth area of computing.

Why should this be so?

Quite simply because the traditional media is not detailed enough or fast enough with the information we require.

Sometimes that media can't even begin to cope with the things that can be achieved with these boards - like providing home based education for isolated kids, or getting fast help for specific problems.

So it is appropriate that this month we take a look at CoCo's place in all this and see what can be achieved with the CoCo and a little Telecom phone line.

*

If you have been thinking of purchasing a modem, now is a good time, with the release of several cheap modems which can handle both the 300 baud and 1200/75 protocols.

In fact Paris Radio have a nice package deal at present, as do Tandy.

*

Last month we spoke about what we are doing with CoCoLink, and how we will be moving our operation eventually to Minerva.

Thank you to those who have already contacted us on that system. By late January, we'll be there in force!

*

The people who make this magazine what it is, are folk like Robert Webb of Tamworth.

Robert is more typical of our contributors, than one might at first suppose.

There are other users in Tamworth, but until the latest Tandy guy Jim Patching arrived, there has not been a lot of support for the local user from Tandy in that city.

Robert battles on, on his own and gets involved in some pretty heavy stuff.

The result is some excellent material contributed to CoCoOz and Australian CoCo over the years.

Robert also backs up users over a much broader area, and helps the folk in Armidale and Gunnedah (I think).

Now Robert is starting the first Tandy Robotics Special Interest Group. This is especially interesting, in the light of what I have said, because the other group which is big on Robotics is at Wagga Wagga!

As readers will be aware, I want to be involved too.

I have two major personal interests to feed, the first is my love of trains and model trains - and as we are discovering, the CoCo is really over powered when it comes to doing that!

The second interest is from the educational standpoint.

For too long, we have allowed our kids' school computing time to be wasted with ill conceived games and exercises which the teacher would normally have provided with traditional means in a much more effective manner.

The job of the computer in the school needs to be defined, and we need to find tasks for it to do which are meaningful, and which really to assist the student in his/her attempts to come to grips with the real world.

Enter Robotics and other associated external interfacing.

Here is where we can make a contribution in the schools, where kids can obtain information and proof of otherwise bland theories, and where a school can undertake projects which can have a meaningful impact on the surrounding community.

So we applaud Robert, and we look forward to hearing of his progress.

The Computer Hut in Bowen has taken over the distribution of most of the software that Software Spectrum used to sell. Tony Evans from Computer hut also has a number of additional lines, including some very good Australian items.

Tandy had a 16 page software list from this company which they distributed in December. If you missed out on one, see if your local Tandy shop can get you one.

*

Rumours Dept. If you like dabbling in what might be, here is the latest word on 'the new CoCo'.

It is said to be either a 512K or 640K 68000, with OS-9 level 2 as the native operating system. (See article this magazine for details on this system.)

RS Dos will load as a task, so you will be able to run most of your existing CoCo software.

The Tandy Hard Disks are to be made available for the existing CoCo, so it is assumed that these will be an important, but not necessary part of the 'new CoCo'. The story goes that otherwise, we may be using 3" drives on the 'new CoCo'.

To quote Donald Sutherland in 'Kelly's Heros', using OS-9 level 2 with a 68000 CPU, "would give you a very nice edge"! (Donald was fighting the Second World War at the time, and his edge was a Sherman Tank which he wanted to use on a bank job - but the parallel is pretty close!)

Well we are into the new year, and once more Christmas is behind us.

For those of us with kids, this time of the year gives us an opportunity to relive our younger days, and touch again the innocence lost with the passing years.

The new year is the traditional time chosen for changes to formats in magazines, but apart from a little 'trimming' here and there, we don't expect to be making many changes to Australian Rainbow - if the Australian dollar can stay above $0.68 US!

The worry we had all last year is still with us as we begin this year. It is a concern which makes Tandy and the other suppliers of imported goods most uncomfortable too. Any change to the value of the dollar will have to be reflected in a pricing change - a situation none of us wants to see!

*

During January ONLY, as a special offer to get you all relaxed and ready for a great year of computing, we will supply you with the two parts of the Best of CoCoOz #2 (Games) for $13.20 on tape , and $37.00 on disk.

*

It looks like there may be THREE CoCoConf's this year!

"Small" Conf's will be held in Melbourne and Sydney around April. Then a bigger Conf is planned for the Gold Coast in about August.

Do plan to be at one of these. CoCo is NOT dead - our little computer is more capable than ever, thanks to the great software available now. But like everthing else, it requires that you invest some of YOUR time and energy to get something out in return!

The other reason for being at a conference is that owning a CoCo admits you to a "family" which is quickly becoming an institution. I personally look forward to the conferences, just so that I can meet and talk with you.

*

We've given most of our article writers a holiday this issue, so some of the regular columns are missing, but look forward to some great reading this coming year!

The Forth boys have got their heads together to provide a series of articles on Forth that even I can understand!

Kevin has a very interesting series of articles for new users of Assembly Language.

Jack Fricker is preparing some articles for OS-9 users.

Our Education column is going to be full - already we have some excellent stuff from Bob Horne, and the Delbourgos.

The Delbourgos have a new column which will support their Extended Colour Basic package.

Geoff Fiala and I will be continuing the series on how to use the CoCoConnection.

And surrounding all this information will be more advanced programs for your CoCo. Programs which are not only different, but which serve to teach you more about your computer!

Have a great year, and keep sending your work - as you learn, others learn from you.

Software Review

Protector Helps Eliminate Program Piracy

Software piracy. This doesn't sound like such a bad thing. Actually, it sounds kind of exciting — makes you want to buy an eye patch and a parrot. After all, you are not actually stealing anything, you are just "copying" your buddy's program for your own use, or trading a few programs with your friends. What possible harm can that cause?

As a person who has marketed Color Computer software, I know better. Many hours are spent writing, testing and debugging software. If the author does not receive what he considers adequate compensation for his effort, he won't create another program. It's just not worth it.

The programs I sold were written in BASIC, were not copy protected and were designed to be used with a speech pak. They sold for a reasonable $9.95. When they were sold at a Southern California RAINBOWfest, often a group would come to the booth, examine the programs and buy one copy, split the cost and copy the program for everyone later. Since there was no copy protection, duplication was as easy as typing CSAVE.

Six generations of "trading" from only one sale with four copies being made from each copy generates 2,625 copies of a program. The market then becomes saturated and the product dies.

My reaction to seeing so many unsold copies circulating, after an initial rise in my blood pressure, was to forget about the software market. It was just not worth the expense and effort. If enough authors walk away from the Color Computer market, there won't be any really good software available for our computer.

An alternative to quitting the business is being offered by Racine Software to authors of cassette-based BASIC programs. Called *Protector*, it prevents copying of BASIC programs by ordinary means.

After loading and executing *Protector*, load the BASIC program and resave the modified program for sale. The customer can then no longer use the CLOAD, CLOADM, CSAVE, CSAVEM, PEEK, NEW, LLIST, SKIPF, DEL, TRON, DLOAD, RENUM, POKE or LIST commands. The BASIC program is also loaded in sections, like an ASCII-saved program, to the accompaniment of clicking cassette relays and varying volume levels that will negate the various cassette program copiers being offered. These protection devices are combined to take the "yo-ho-ho" out of some software pirates.

Of course, there are means of bypassing *Protector*. Since this program generates a cassette, it could be copied via dual audio cassettes, but the reliability and consistency of an audio copy is poor.

It's a shame that programs such as *Protector* have to be offered, but it's a better alternative than no programs at all.

While *Protector* is a fine, functional program, the packaging (none), documentation (four somewhat confusing xeroxed sheets loaded with typing errors) and the quality of the cassette (K-Mart C-60) leave quite a bit to be desired. A program of this quality deserves better packaging and documentation.

If you do, or are intending to market cassette-based programs for the CoCo, the salability of your program will be extended by using *Protector*.

Software Review

Soccer Statistics Package Keeps Stats In Check

Soccer is enjoying increased popularity here in the States, and Sugar Software is out to provide assistance to soccer coaches and fans with a new statistical package designed to maintain accurate records of teams and their opponents. The program is written in Extended Color BASIC and requires a minimum of 32K, one disk drive and a printer.

Having reviewed other Sugar Software programs, I can honestly say they continue to produce well-written and user friendly programs for the CoCo. *Soccer Statistics Package* is completely menu-driven, simple to use and comes with a thorough 13-page instruction booklet. However, it is so self-prompting that once you begin using it, you will find very little need to refer to the instructions. Also included is a set of sample data for the user to "play with" to get the feel of the program.

The main menu consists of the following options:

1) Player stats new start
2) Player stats continued
3) Opposing stats new start
4) Opposing stats continued
5) Review or correct
6) Add new players
7) Output stats to printer
8) Print team summary
9) Goalie stats
10) End program

Options 1 and 3 are used at the beginning of the season to load initial information. Options 2 and 4 are used during the season to enter continuing statistics. Option 5 allows you to display and correct existing information. Options 6, 7 and 8 are fairly straightforward and option 9 allows update of the goalie stats, which, because of their different content, are maintained separately.

Once the statistics are loaded, they can be printed in several different ways:

1) A basic printout of all statistics entered.
2) An individual player summary by game of each player's efforts, including scoring percentages.
3) A team summary containing all accumulated stats to date for all players.
4) An opposing team summary by individual team with a column for total points scored to date, scoring percentage by opposition to date and average points by opposing team to date.
5) A printout of the goalie stats for each goalie by game with the percent of opposing shots scored on each

goalie.

6) A goalie summary containing accumulated totals of all games played to date for each goalie.
7) Finally, since all reports contain abbreviations for the column headings for each statistic, an optional, single page explanation of each abbreviation can be printed for ease of reference.

I hope this gives an idea of what this package is capable of and whether or not it could be of benefit to you as a coach, player or fan. If you are a player, perhaps your coach would be interested in your assistance in computerizing the team's records. Whatever your decision, this is a well-done program that deserves serious consideration.

— Ken Boyle

Software Review

Colorcom/E Is An Excellent Smart Terminal Package

Disk *Colorcom/E* is a complete smart terminal package designed to work with a 32 or 64K Color Computer. And smart it is!

This latest Version 3 supports XMODEM file transfer protocol so you can download machine language programs. This is an excellent feature and a real time saver when downloading such files as *Graphicom* pictures from CompuServe, where every minute of online time counts.

The package consists of two diskettes and two well-documented booklets. The booklets deal with the operation of the program, complete with examples on how to get started and many examples of command sequences and the expected results. The instructions are well-written and easy to understand, however, the section dealing with the Auto-File log to Spectrum Projects Bulletin Board did not work! This was later found to be the result of this BBS no longer being in service. Although this was a minor problem, the user who tries to call that BBS will encounter the same fate. (Bob Rosen informed me that he will soon reconnect his system.)

The Edit Auto File mode is easy to use. I simply added the phone number and logon sequence, then called a local BBS with no problem. This Auto File mode is also a real time saver since it allows all the repetitive typing needed to logon a particular BBS up to the CoCo. If you're fortunate enough to own a modem with auto-dial capability, then so much the better.

The program can be customized to keep trying the number until connection is made. That feature, coupled with the Wake-Up mode, will knock your socks off. You can program the time of day you wish CoCo to start dialing the number — now that's power! The manual specifies that the Wake-Up mode be used only with well-tested Auto-Files for obvious reasons, however, a Set Master Timeout mode is provided that will disconnect the modem if an expected result does not occur within a reasonable time.

In the command mode the screen is split. The top portion displays data stored in the buffer while the bottom portion displays a menu of available commands. To examine received data, arrow keys are used to scroll forward and back in the file.

SHIFT arrow keys are used to display groups of lines in the file, depending on whether or not you are in the 32- or 51-character per line mode. Hitting 'T' will show the oldest data in the buffer and, if you wish, a particular section can be selected to send to your printer or save to disk. The 51-character display is easy to read, and is sharp and clear on my amber monitor.

The usual disk commands are also provided. You can (R)ead from or (W)rite to disk as well as perform a (D)irectory.

Most parameters of *Colorcom/E* can be changed to allow virtual compatibility with any BBS and computer system:

— RS-232 modem and printer protocol
— Automatic capture of characters used to open or close your buffer
— Word mode to prevent on-screen word splits
— XON/OFF for uploading files
— Carrier detection for modems without this feature
— Set time between characters when sending files from the transmit buffer. (This is valuable when the host computer cannot accept data at full speed.)

In addition to all of *Colorcom/E*'s neat features, one of my favorites is a utility called "DFT." This Direct File Transfer allows rapid, error-free file transfer between two CoCos that are running *Colorcom/E*. DFT automatically transfers the file and corrects errors detected during transmission or reception. After the file is received, it is automatically saved to the receiver's disk using the same filename that is transmitted.

DFT is menu driven and very easy to use. I used it several times to transfer programs and never had a problem. Even if someone in your house picks up a telephone extension and makes noise on the phone, DFT will detect the error and retransmit the block of data until it's received correctly.

While *Colorcom/E* is copy protected, two identical diskettes are provided so one can be stored as a backup. It also has a 90-day warranty by Spectrum Projects. An owner's registration card is provided that can be mailed to Eigen Systems, which puts you on their mailing list for future patches or updates.

In summary, Disk *Colorcom/E* offers a great bargain. It's an excellent product that is easy to use and provides the user with all he or she needs to communicate with the many bulletin boards available.

—Jerry Semones

Software Review

With *LABEL64*, It's Listing 1-2-3

Do you dread the holiday season's approach because of all of the cards you'll have to mail? Maybe it's time to reconsider just who your friends actually are, or maybe it's time for an address label program. *LABEL64* is a simple-to-use label program that is good for keeping Christmas card lists, club lists or any other long list of addresses.

LABEL64 comes on cassette and is designed for a 64K machine (it warns of problems with lesser amounts of memory). The three pages of documentation thoroughly explain the eight functions: input names, delete names, clear memory, locate names, save to tape, load from tape, list to screen, print labels and sort. One of the main attractions of *LABEL64* is that it is capable of filing, sorting and printing 300 records (addresses) at one time. It takes advantage of practically all of the computer's memory, leaving nearly 32K after loading. While in use, mistakes are easily corrected with clear prompting, and sorting is a snap.

Being an associate pastor for a church, I frequently need to send out mailings to the kids in my youth group. I keyed in the membership, then found my first criticism with the program. The addresses, as printed, were not configured to the three-across labels I had purchased. I attempted to correct the program myself, but was unable to. A quick call to Owls Nest Software put me in touch with Norman Shelton, who ably led me through not only the new parameters I needed, but also an explanation of how I could save the updated machine language version to tape.

Although it doesn't claim to be a database program, I found myself wishing for some way of being selective about the addresses I printed. The two options available for printing are all or one at a time. Finally, I found it cumbersome to have to use tape rather than disk. In all fairness, though I realize it would use up a good bit of memory, the ease of use would, I believe, far outweigh the memory loss. All in all, *LABEL64* does print labels just as it claims to do.

— **Jefferson L. Hatch**

Software Review

War of the Worlds: Fast-Paced Adventure With Good Graphics

Most Adventures start off with a phrase something like this: " . . . You are in a field. In the distance you see a castle. Possible directions are North, East, West, and South. . ." at which point the novice Adventurer nervously has a quick drink from his handy glass of soda or coffee, wrings his hands, licks his lips and pounces on the keyboard with heart pounding.

The veteran Adventurer, cool, calm and ready for anything at this stage, smiles quietly at the familiarity and commences with the usual routine of looking, inventory and so forth, as practiced fingers efficiently tap in the single letter abbreviations of commands and sentences that are a terse two words.

War of the Worlds is different. From the start, you are greeted not with the scenario, but a screen called "Master Control," which is available at any point in the Adventure by pressing the space bar. This is the only contact needed with the keyboard. Master Control gives options to quit from the game, save a game, load a previously saved game, start from the beginning or resume play. Choices are made by using the right joystick to place a rectangle over the option.

Assuming you are starting a new game, the title and a text screen giving the background of the Adventure appears, then surprise number two arrives: graphics! The graphics are black on a green background, yet they do add a sense of detail and realism to the Adventure. They are fairly well-detailed, clear drawings that resemble a series of lithographs. I found them to be more of an enhancement than a detraction to the Adventure. Certainly, color graphics could add to the program substantially, but there is a problem with enough memory to store the color as well as the drawings. Therefore, Triad Pictures decided on highly detailed graphics instead of color.

War of the Worlds is an Adventure in three parts, called "chapters." Chapter One is titled "The Landing," Two is "The Quest " and Three, "The Last Hope." It is advisable to play the chapters in order since they do follow a plot.

This Adventure is for the moderate to advanced Adventurer. Although it is entirely joystick-driven, and the options for each of more than 200 situations per chapter are right on the screen, this is a tough one to solve. As an experienced Adventurer, I have not entirely solved all three chapters as of this writing, so fear not, I will tell no secrets! I will, however, suggest that if you are a novice looking to sink your teeth into a tough one, this is an excellent Adventure on which to start. Those who are experienced, dive right in; this one takes time and careful thought!

War of the Worlds comes with an eight-page booklet that gives loading and game play instructions as well as some background to the Adventure, which is refreshingly thorough considering most Adventures rarely give more than a paragraph or two of documentation. There are also two maps in the manual. What they are and how to use them is for the Adventurer to find out!

One minor omission is the failure to mention use of a mouse. The manual specifies a Radio Shack-type joystick, but I found a mouse to be much easier and faster to work with. The manual also states that the program may not work with the disk drive controller plugged in, but I found the program to function without difficulty with or without the controller.

Because the scenario of *War of the Worlds* is that of an alien invasion of Earth, it is particularly easy to get "killed," so caution is always necessary in dealing with these unpredictable creatures. Also, don't underestimate the Priest — he may seem like a sluggard at first, but he comes up with some lifesaving ideas from time to time!

This is a moderately difficult, well-written Adventure. It is unique in its combination of graphics and joystick control. Because it is written in three 64K chapters, it provides a more sophisticated challenge to the player. Despite the few minor drawbacks mentioned, I recommend *War of the Worlds* to anyone with a healthy desire to explore the unknown.

— **Jeffrey S. Parker**

Software Review

Naugus Provides Hours Of Enjoyment For Adventurers

When I first received *Naugus* to review, I thought, "Oh boy, a game! This looks like fun." I settled down for an evening and began by reading the instruction manual. While reading, I found myself lost in a fairy tale world. A knight in shining armor is betrayed by circumstances beyond his control. He has to fight terrible monsters to find a magical scepter to redeem his honor.

When I started playing the game, I realized right away it was not a run-of-the-mill game that can be mastered in one evening. But after a couple of nights of playing, I finally felt comfortable enough to write about it (even though I've never been able to obtain the magic sword).

Naugus is a machine language Adventure game for people aged 12 and up. The Adventure begins before you actually start the game. In the instruction manual, which is quite fun reading, it tells us that Byron Axehead, captain of an infantry troop for the king, gets involved in an altercation in a local tavern where a patron is killed by Axehead's sword. This patron happens to be the brother of the king, which means Axehead will have to be beheaded. Since he has been loyal as a captain in fighting on the northern frontier, he is given a chance to redeem himself. He must obtain and return to the king a magic scepter hidden in the magical forest, which is protected by the Naugus. Before he can get to the forest he has to work his way through two other forests that are inhabited by Zombies and Lyxes. He is not allowed to take anything with him, but must find everything he needs in the forests. In this game, you are Byron Axehead.

The game itself consists of three forests you must work your way through. Each forest has 25 screens, which are moved through by using the right joystick. By pushing the number '1', you can see a map of all 25 screens in the forest and which screens you have been through.

When first starting *Naugus*, you enter the first forest protected by Zombies. They are rather slow creatures who have one redeeming quality: They don't leave any bones around if they kill you. There are two things that must be done in the first two forests: 1) Find the key and door into the next forest, and 2) Gather all the weapons and tools you can find.

The weapons are a sword, which can only be used at close distance, and a longbow, used for long distance protection. To use the longbow, you must have quivers of arrows. Each quiver has five arrows, so you want to gather as many quivers as possible. Running out of arrows or not having a sword can be very harmful and shorten your life significantly.

There are also tools you can gather: a rope to climb trees (which is helpful since Zombies and Lyxes cannot), a boat to cross water, magic potions and healing potions to restore strength. At the top of each screen is a display that shows what's in your possesion.

The second forest is protected by the Lyxes. These creatures have no redeeming qualities. They will chase you and throw things — formidable opponents to say the least. You can also gather weapons and tools in this forest.

The third forest is protected by the Naugus who will pursue you from screen to screen. (In the first two forests, Zombies and Lyxes can only move within the screen on which they are found.) The Naugus cannot be killed like the Zombies and Lyxes; it can only be stunned. After obtaining the scepter you have to escape the Naugus and get out of the forest. This is a challenge for even Byron Axehead, former captain of the king's Border Legion.

I found *Naugus* very challenging. It's not the kind of game you can master in one day. The graphics and sound are good and utilize the capabilities of the Color Computer. The manual that comes with the program is brief but very entertaining, while it provides the essential information to get you started. I think *Naugus* is a good cross between arcade-type games and Adventure games; it uses both the manual dexterity skills necessary for survival against enemies and the deductive skills needed to find solutions and develop strategies. I recommend *Naugus* to anyone who wants to have many hours of enjoyment because it is difficult, yet can be won.

— Thomas E. Nedreberg

Software Review

DynaCalc OS-9 — A New Horizon For Spreadsheets

DynaCalc is *the* spreadsheet for the Color Computer and is the only program on the market that spreads over the entire Color Computer operating environment. At first, it was only available for the FLEX operating system. Then came the Radio Shack Disk BASIC version. Now the OS-9 version has arrived. What other program can boast such a claim?

The FLEX version was reviewed by Dan Downard in the August 1983 issue of THE RAINBOW (Page 158). The Radio Shack Disk BASIC version was also reviewed by Dan Downard and appeared in the September 1984 issue (Page 219). All of the commands and functions were included in these reviews. Therefore, I will limit this review to how *DynaCalc* operates under OS-9.

The *DynaCalc OS-9* is supplied on a standard OS-9 formatted diskette. It is not copy protected (you are told to make backups and not to use the original). The required equipment includes a 64K Color Computer and at least one disk drive. Optional equipment includes a printer, joystick or mouse and the OS-9 operating system. That's right, OS-9 is optional. The reason for this is that on the diskette there is already a BASIC OS-9 version 01.01.00 operating system. However, this diskette includes only a small subset (four or five) of OS-9 commands.

Once you boot *DynaCalc OS-9*, the OS-9 operating system is all but transparent. During normal operation, the only reminder of OS-9 is when you save or load a file. It asks for a pathname instead of a filename. You do not have to know all of the little tricks of OS-9 in order to fully benefit from the power of *DynaCalc OS-9*.

There is about 12.3K of work space with the HELP function activated. There is close to 21K if this function is deleted. The manual states that with 21K, there is enough work space for 1,750 numeric cells. As with any OS-9 application, if you practice memory conservation, this will generally be enough work space.

This version operates almost exactly like the other versions in performance. The command format is identical and the commands are the same. It has the standard *DynaCalc* Hi-Res screen, however, it does not have the graphics printing capability of the Disk BASIC version. *DynaCalc OS-9* will drive any printer connected through the RS-232 serial port.

The program itself is excellent. The instruction manual is the weakest part of the package. In the "Getting Started" section, instructions for booting OS-9 are the only ones given. You are not told how to boot *DynaCalc* until the very last page of the manual. In order to boot it, you must type DYNACALC and ENTER at the OS-9: prompt. The major flaw in the manual is the actual construction of it. Over the month that I have had to review this product, one page has completely fallen out of the manual with several others ready to follow. Except for these drawbacks, the manual is very informative.

I highly recommend any version of *DynaCalc*. If you are running OS-9 or are interested in OS-9 and need a spreadsheet, then *DynaCalc OS-9* is *the* spreadsheet. OS-9 needs serious business applications; this is a giant leap in the right direction. (Caution: Handle the manual with care!)

— John R. Curl

Software Review

Utility Routines Contains 28 Subroutines In One Package

By A. Buddy Hogan

Utility Routines is really misnamed. If anything, this neat little package should be labeled "Utility Bonanza." It contains 28 of the most useful machine language subroutines I have come across — all in one neat package.

For the very new CoCo users, a subroutine is part of a program called upon to perform some task that is used by the main program. A program may use as many subroutines as the programmer needs, within the limits of available memory. A subroutine is ordinarily "called" by the main program by EXECuting it or by a GOSUB command. Some programs use subroutines contained in the CoCo operating system (Color BASIC, Extended Color BASIC or Disk Extended Color BASIC) to perform certain tasks. In short, subroutines are *included* in a program, they are not stand-alone programs.

The following subroutines are contained in the package from Microcom Software (the number of bytes used by the subroutine and the CoCo system requirements are at the end of each description):

1) Border Creation — Creates as many as 255 different border styles on the text screen. These borders consist of letters, symbols and graphics symbols. (35 bytes/16K, ECB)

2) CLEAR-Key Disable — Disables the CLEAR key during INPUT/LINEINPUT commands so you won't accidentally erase the screen. (39 bytes/16K, ECB)

3) Command Keys — This powerful routine allows you to use two keystrokes (the down-arrow key with any of the 26 letters of the alphabet) to automatically enter many common CoCo programming terms. This is useful for entering long program listings. (254 bytes/16K, ECB)

4) Cursor Styles — Create up to 65,000 different steady or blinking cursors. (92 bytes/16K, ECB)

5 and 6) Error Skip — Two routines are provided, one for BASIC and one for machine language programs. This routine is usually called "ON ERROR GOTO." The necessary line to be included in your BASIC program is given in the manual for the actual ON ERROR GOTO instruction. (47 bytes (BASIC)/16K, Color BASIC; 58 bytes(ML)/16K, ECB)

7) Full Length Errors — Print out the error message in full instead of the two-letter abbreviation. (762 bytes/16K, Color BASIC)

8) Graphics Scroll — Allows you to scroll up the graphics screen in any PMODE. (51 bytes/16K, ECB)

9) Key Clicker — Every time a key is pressed, you will hear a click. This helps you type with the confidence of knowing a key has been "recognized" by the program. (39 bytes/16K, ECB)

10) Line Auto Increment — Usually called "Auto Line Numbering," this utility automatically puts in line numbers when you are typing in a program. (181 bytes/16K, Color BASIC)

11) Memory Detection (32K/64K) — This routine answers the questions, "Is it 32K or 64K?" and "Was the memory upgrade installed properly?" (33 bytes/32K, ECB)

12) Pause Control — Allows use of the SHIFT key and space bar to pause any BASIC program and some machine language programs, and the SHIFT key and space bar to resume the action. Lets you have your cake and eat it, too: play computer games and answer the door, eat, etc., without losing the results of hours of computer time. (101 bytes/16K, Color BASIC)

13) Repeat Key — Not only will this routine allow you to have the key being pressed repeated as long as you hold both it and the CLEAR key down, you also can choose up to five repeat speeds. (116 bytes/16K, ECB)

14) Reverse Video (Green) — Changes the bright green screen display to a more pleasing "soft" green that is easier on the eyes. (208 bytes/16K, ECB)

15) Reverse Video (Red) — Makes white characters on a red screen. (217 bytes/16K, ECB)

16) Save Graphics to Tape — Saves a graphics screen to tape. You may then load the graphics screen back from the cassette whenever you want. (22 bytes/16K, ECB)

17) Save Text Screen to Tape — This one saves a text screen to tape. (24 bytes/16K, ECB)

18) Screen-to-Printer — Whenever a character is printed on the screen, this routine sends it to the printer, also. (71 bytes/16K, ECB)

19, 20 and 21) Spooler (16K), Spooler (32K) and Spooler (64K) — These utilities allow you to continue using the computer while you're simultaneously using the printer. They set up a buffer of 2K (Spooler 16K), 4K (Spooler 32K) or 32K (Spooler 64K) to handle the material to be printed while you do something else. (2,229 bytes (16K), 4,277 bytes (32K), 184 bytes(64K)/16K, ECB)

22) Super Error — Places you in the EDIT mode whenever you encounter an error in program execution. (86 bytes/16K, ECB)

23) Super Scroller — This routine really lives up to the name "utility," meaning useful. It will save up to 32K of screen text and lets you scroll through it forward or backwards. It even beeps when you get to the end of the buffer. The line saving feature may be turned on or off. Now

you can list that 1,000-line program and scroll through it at your leisure. (355 bytes/64K, ECB)

24) Tape Index System — Creates an index of all the files on a cassette tape. It skips all I/O Errors so you can easily index "bad" tapes. The index includes the file type (BASIC, machine language, data) and the file format (ASCII or binary). (140 bytes/16K, ECB)

25 and 26) Tape to Disk Copy — Comes in both 1.0 and 1.1 Disk BASIC ROM versions. It will transfer BASIC and machine language programs from tape to disk. The manual claims the 1.1 ROM version will handle 95 percent of the 1.1. BASIC and ML programs. (228 bytes (1.0) and (1.1)/16K, Disk BASIC)

27) Text Screen Scroll Protect — Keeps the text screen from scrolling. (123 bytes/16K, ECB)

There is another routine, called "Console In Routines," that is a combination of the Repeat Key, Cursor Styles, CLEAR Key Disable and Key Clicker routines all in one. These routines are combined because the manual lets you know ahead of time that it is very difficult to use any of these routines together in your program, so it has been done for you.

The joy of this review was not just the availability in one place of most of the utility routines I have needed or read about, but, most of all, the willingness of the author to share with the purchaser all of the details about the routines' operation and how they may be used in one's own programs.

The routines are explained so even a novice machine language programmer can understand what the author is doing. Each routine is listed in full and the manual even includes a program that will allow those who don't have an editor/assembler to key in the listings. The package would actually make a good machine language tutorial workbook for use in the classroom or by individuals trying to learn machine language programming on their own.

The manual is bound and consists of 80 pages (8½ by 11). The author, Kishore M. Santwani, advertises on the cover that it contains "Routines that will otherwise take years of programming experience to learn." That is an understatement.

If you don't want to key in all these routines, you may purchase them on cassette or disk. The manual that comes with the tape and disk is an abbreviated 13 pages (8½ by 5½) stapled in the middle. It contains only the essentials, with no program listings or detailed explanations — just what it will do and how to use it.

Microcom Software has certainly made a valuable contribution to the CoCo Community with *Utility Routines*. They have even gone the extra mile to allow software authors to use these routines in their programs without the need to pay any royalties. You can pay plenty for some of these routines, which are sold separately by others, or you can get them all in *Utility Routines* for peanuts.

Correction

"Screen Pokes Made Easy" (December ,1985 Page 45): Bill Bernico has informed us that there are two minor errors in the text. The listing referred to as *SCRNPOKE* in the text should actually be called *TITLPOKE*; the reference mode to Listing 1 should really be Listing 2 (*DEMOPOKE*).

Accessory Review

No-Stat II Protects CoCo And Eliminates Static

Jutta and I share a CoCo here at RAINBOW. She calls him "Cookie" and what I call him can't be printed — he likes her best; always has. But Cookie has been a lot friendlier lately since I gave him the No-Stat II from Amber Enterprises.

No-Stat II is an anti-static pad made of inert conductive foam which is laminated to a tufted fabric. It is flame retardant and non-allergenic. Measuring 11 by 15 inches, it collects a lot of dust that sooner or later might endanger Cookie's sensitive IC chips. (I know of these only through Tony DiStefano's column, but I have a lot of respect for them.) No-Stat II has also stopped the nasty little zingers our tough Cookie packs for me when I shuffle over the carpet to his keyboard instead of picking up my feet.

I'm glad I gave No-Stat II to Cookie because the "CoCo Gallery" entries I review on him each month are important to me as well as a source of delight to the entire office, and I need Cookie to cooperate. He seems to be behaving better toward me and I can only credit that fact to No-Stat II (it's the only present I've ever given him).

No-Stat II pads also come in a 28 by 28-inch size, and I'm thinking of ordering one for "Sulu," the Tandy 1200 HD I use. (He's *my* friend!)

— **Monica Dorth**

INFORMATION EXPRESS

Copyright 1985, AgriData Australia Pty Ltd.

(I wish to thank Ian Blair for giving me the information I needed to do this review. Ian did the first draft of this review and I took bits out and put bits in.

I work at Universal Services, Dalby, Qld, (Tandy dealer 9407). I have sold a number of subscripions to Information Express to people who have said that the information they get out of Information Express is the same as the get out of the news papers etc but it takes less time and phone calls to get. Andrew Simpson)

WOULD THE RIGHT ANSWERS AT THE RIGHT TIME
MAKE YOU GREATER PROFITS ?

In today's volatile markets, better-informed business and investment decisions are based on accurate, current information.

Until now, the primary sources of market and price information and advice have been the traditional media. Frequently such sources aren't as immediate or accessible as you may require.

Today the information you need can be instantly available. Anyone equipped with a personal computer, data terminal or communicating electronic typewriter now can be connected promptly by telephone line to a dial-up computer database.

Information Express is Australia's first and most extensive dial-up database for reports and advice on share prices, currencies, money market, commodities, news, weather, and agribusiness.

Information Express makes use of state-of-the-art computer and communications technology, but is essentially a publishing - not a computing - business. It is owned by a group of leading Australian publishers, including The Herald Times, Queensland Press and Advertiser Newspapers, plus Techcom Australia, a pacesetter in the field of electronics communications.

INSTANT PROFIT-MAKING INFORMATION

At Information Express they concentrate on providing perishable, rapidly changing information of the kind generally required by business people, who make frequent buying and selling decisions.

The share market is an obvious example.

Information Express offers a service in many ways as comprehensive as the powerful real-time systems used by brokers. It provides the latest price for all shares and options quoted on the Melbourne and Sydney Exchanges within seconds, but it is not updated as you watch. Each access is a snap-shot of the prices on the Exchanges at that moment, not a movie.

Information Express costs as little as one-twentieth the price of the real-time systems.

For people who are not brokers, it is the best way to obtain low-cost, up-to-the-second access to share and option quotes. As well as share prices, Information Express has huge databases of money markets, futures, currency, equity and agricultural information. Overseas reports from leading sources are kept up to the minute by satellite connection.

AGRICULTURAL/RURAL

(1) Livestock Market Reports: Daily from Dalgety Farmers Ltd., John Small, Livestock & Meat Authority of Qld., and State Departments of Agriculture.

(2) Grain & oilseed Prices: Daily and weekly from NSW Grain sorghum Board, NSW oilseeds Board, Continental Grain Australia and other sources.

(3) Cotton prices: Daily from the New York Cotton Exchange, and Commodity News Services (CNS).

(4) Wool Market reports: Daily from the Australin Council of Wool Buyers and the Australian Wool Corporation.

(5) Gross Margin Analysis: By district and enterprise from the National Australia Bank.

(6) Trends and Forecasts / World Supply and Demand: Bureau of Agricultural Economics, United States Department of Agriculture, Graintrends, Helming, Doanes, and CNS.

(7) Taxation: Depreciation rates, allowances, averaging, sales tax, exemptions, from Pannell, Kerr, Foster; and Michael Boyce & Co.

NEWS/WEATHER.

Local and International News:

As it happens from leading newspapers around Australia and their international sources. Including the Melbourne Sun and Herald, The Weekly Times, Queensland's Courier Mail and Sunday Mail, the Adelaide Advertiser, and the West Australian. Also Power Farming, Ag Dealer, Primary Industry Newsletter, and KRF. All searchable by Keyword.

Weather Forecasts:

3 times daily from Bureau of Meteorology for all districts of Australia. Worldwide reports and crop weather from Globex Weather Services. Weather alerts from the Bureau of Meteorology.

LOW-COST MESSAGING AND TELEX.

In addition to up-to-the-second reports, Information Express has a powerful but easy to use electronic messaging capability. The messaging service is interconnected with the telex network worldwide. This service permits you to send or receive telexes via your own terminal and avoid the cost of a telex machine. And because you can purchase a portable terminal, you can be in touch around the clock wherever you are.

OBJECTIVE INFORMATION SOURCES

Information Express is completely objective - they are neither dealers nor traders. They source the most respected companies to provide our service with information.

THE VITAL INFORMATION AVAILABLE ON THE INFORMATION EXPRESS DATA BASE.

Business/Finance

Share Prices and Volumes: On line as they happen from the Sydney and Melbourne Stock Exchanges. Daily summaries on Australian and overseas markets from Jacksons Limited, Potter Partners and McCaughan Dyson. Detailed U.S. information from Knight Ridder Financial (KRF).

Money Market and Interest Rates: Daily closing rates in the U.S., plus domestic money market and bank bill rates from the ANZ Bank. Also, U.S. Bank Acceptances, Commercial Paper, Certificates of Deposit and more, plus commentaries.

Currency and Foreign Exchange: Austrailian rates updated four times daily from the ANZ for all major currencies, as well as international currency markets updated continously from KRF. Daily reviews from New York, London, Singapore, Hong Kong and Tokyo.

Futures and Options: Real-time quotes from the Sydney Futures Exchange. Overseas information updated with a 10 minutes delay during trading hours (COMEX, CME, LME and others).

Commodities:

Daily Prices and Commentaries: From local and overseas sources. Prices are updated with a 10 minute delay during trading hours from the London Metal exchange (LME), London Commodity Exchange (LCE), New York Metal Exchange (NYMEX), New York Commodity Exchange (COMEX), the Chicago Board of Trade (CBT), and Mercantile Exchange, Kansas' CBT, New York's Cotton Exchange, Minneapolis' Grain Exchange, Winnipeg's Commodity Exchange, Kansas City's Board of Trade, Coffee, Sugar and Cocoa Exchange.

Local commentaries from Jacksons Limited as well as overseas reports from United States Department of Agriculture (U.S.D.A.) and Knight Ridder Financial (KRF).

Precious Metals: Gold, Silver, Platinum, and Palladium.

Base Metals: Lead, Zinc, Aluminium, Tin, Nickel, and Copper.
Agricultural: Grains, Soyabeans, Cattle, Hogs, Corn, Beef, Cotton, Sorghum, Orange Juice, Cocoa, Potatoes, Coffee, Sugar, Wheat, Rice, Barley, Oats, etc.
Resource/Energy: LPG, Regular gas, Crude Oil, Heating Oil.

Information Express is a publishing service which uses advanced computer technology but is not subservient to it. Your subscription to Information Express provides these advantages:

* You only pay for what you use.
* The information has real value for business people.
* The system is easy to use and connect times are minimised through the use of printed indexes, codefiles, address files, scrolling, updated and keyword searching facilities.
* They provide a comprehensive user manual for training and continuing client services.
* The service operates 24 hours a day, seven days a week.
* Electronic Mail and telexes are available at any time, and are part of the standard service.

HOW DO I JOIN INFORMATION EXPRESS?

All that you need is a terminal or personal computer, a modem and a telephone. A printer is optional, but desirable. You may see Information Express at your Tandy deäler or store. Tandy is selling 6 month subscriptions, so you may try the service or you can go direct to Information Express for a 12 month subscription

The initial fee includes the annual subscription for the service of $399, plus a once-only charge of $40 for the subscriber Manual and Indexes and $86 for connection to the service. In subsequent years, you pay only the annual subscription.

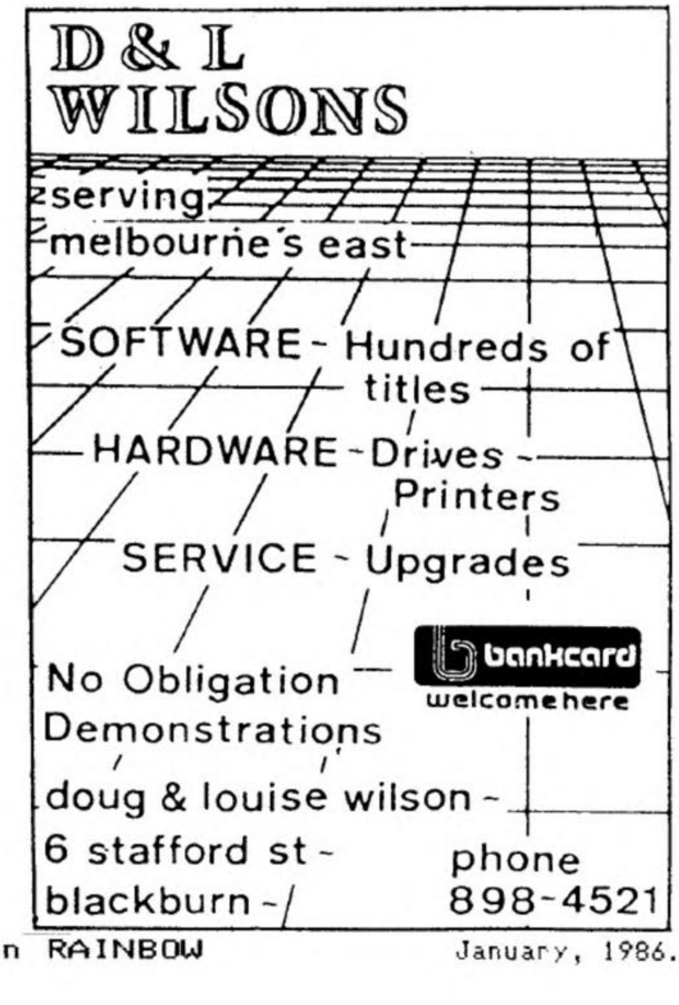
In addition, you pay for connect time to the service and for the reports you access. Typical subcribers pay about $50 a month for these charges. No STD telephone charges are incurred by the subscriber.

Companies or organisations requiring multiple subscriptions can purchase a corporate subscription plan that offers up to ten subscriptions per corporate subscription. The corporate subscription has an annual charge of $1250, with each subscription costing an additional $149 per year.

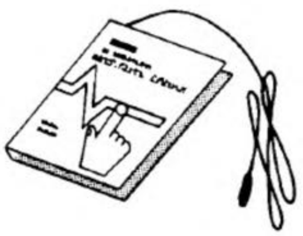
Thus the corporate subscription gives savings with four or more subscriptions and reduces the annual cost per subscription to $274 for ten subscriptions. Each of these subscriptions has its own subscriber number and password and is invoiced separately for corporate accounting purposes.

Information Express is so confident of the benefits provided by their service that, if at the end of two months you are not completely satisfied, they will refund your subscription in full.

Andrew Simpson



Working With The Electronic Book Part 2



By Steve Blyn

This month we present the second of a two-part series on Radio Shack's Electronic Learning Book. Last month, we described how to use this new peripheral; in this installment, we will demonstrate a practical educational application for it.

Just to review briefly, it is our contention that the Electronic Book is a wonderful idea and a pleasure to use with certain students. These include the preschool set and many physically disabled individuals. Both of these groups often have motor dexterity problems and find it difficult to use the computer's keyboard. The Electronic Book offers them an alternative means of entering data. It is also useful for those who thrive and learn best through a new medium.

It was difficult narrowing down to just one subject area to demonstrate the book's use. We feel that Radio Shack often makes the decision to direct their CoCo software almost exclusively to younger children. This decision has caused hoards of overpriced Apples and even TRS-80 Model IIIs to be purchased by middle school classrooms instead of the better and less expensive CoCo.

We felt on safe ground with an arithmetic program. The level chosen is two-digit addition examples. This is often taught in second and third grades. The program can be altered fairly easily to a higher or lower level. Converting to a subtraction or multiplication program would similarly be a simple task. We will explain how to alter the program after discussing how it works.

Lines 110 and 120 tell the computer to check the right joystick port values, JOYSTK(0) and JOYSTK(1). These values are read in and altered by pressure on the Electronic Book's surface. Lines 130-220 read the joystick values to determine if any of the numbers have been pressed. When a number is pressed, it is drawn by Line 250 and a tone is played on Line 260. Since the answers are all two digits, this process is repeated twice by the FOR/NEXT loop created by lines 80 and 300.

Line 310 checks to see whether the child's answer is correct. If correct, a happy tune is played. If incorrect, the real answer is converted to strings and drawn underneath the example by lines 320-370. A pause in the action occurs, then it goes back to Line 70 for the next example.

No scorecard is included in this program as we are concerned mainly with getting our user to become familiar and then comfortable with the apparatus. No timer or limit on the number of examples was included for the same reason. Please feel free to add any of these enhancements as you see fit for your own purposes.

This program can easily be converted to a different degree of difficuly. The numbers in the examples are randomly chosen on Line 640. 'Q' and 'S' are the 10's place digit values. 'R' and 'T' are the unit's place digit values. In our program, we arranged their random values so the sum of the two two-digit numbers does not exceed 98. This means we will always have a two-digit answer. You may change the level of difficulty by altering these values. You would, of course, also adjust the value of YY in Line 80 for the proper number of digits in the answer.

The program can easily be changed for other arithmetic operations. "AN" is the answer the program is looking for. It is computed on Line 660. The answer in our example is AN = (Q*10)+R+(S*10)+T. To change to a subtraction program, for example, change Line 660 to read AN=((Q*10)+R)-((S*10)+T). You should also be careful to keep the value of 'Q' larger than 'S' to avoid negative answers.

Last month, we challenged those readers who also have a Tandy 1000. Although not designed to work on the Tandy 1000, the Computer Island staff found the Electronic Book works fairly well on it. The joystick values are, of course, different. Our challenge was for you to determine those values. The answers are as follows:

AN = 1 if A = 26 and B = 10
AN = 2 if A = 2 and B = 10
AN = 3 if A = 17 and B = 11
AN = 4 if A = 24 and B = 12
AN = 5 if A = 44 and B = 12
AN = 6 if A = 65 and B = 12
AN = 7 if A = 14 and B = 28
AN = 8 if A = 14 and B = 3
AN = 9 if A = 14 and B = 16
AN = 10 if A = 14 and B = 24

The Tandy 1000 also uses the left rather than the right joystick port at the front of the computer. Joystick, incidentally, is called STICK on the Tandy 1000 rather than JOYSTK.

In conclusion, we feel the Electronic Learning Book can be a valuable educational aid to many. We hope we have helped to convince you to pursue this avenue and, also, that Radio Shack continues to support this device. □

(Steve Blyn teaches both exceptional and gifted children, holds two master's degrees and has won awards for the design of programs to aid the handicapped. He owns Computer Island and lives in Staten Island, N.Y.)

```
190 .......27
410 ......168
END .......4
```

The listing: ADDITION

```
1Ø REM"ADDITION WITH THE ELECTRO
NIC BOOK"
2Ø REM"STEVE BLYN,COMPUTER ISLAN
D,NY,1985
3Ø DIM E$(1Ø)
4Ø CLEAR2ØØØ
5Ø FY=RND(-TIMER):REM"RANDOMIZE"
6Ø GOSUB 4ØØ
7Ø GOSUB 59Ø:RA=Ø
8Ø FOR YY= 1 TO 2
9Ø N=Ø
1ØØ X=A:Y=B
11Ø A=JOYSTK(Ø)
12Ø B=JOYSTK(1)
13Ø IF A=63 AND B=Ø THEN N=1:DB$
=E$(1)
14Ø IF A=53 OR A=52 AND B=Ø THEN
 N=2:DB$=E$(2)
15Ø IF A=41OR A=42 AND B=Ø THEN
N=3:DB$=E$(3)
16Ø IF A=8 AND B=Ø THEN N=4:DB$=
E$(4)
17Ø IF A=19 AND B=Ø THEN N=5:DB$
=E$(5)
18Ø IF A=3Ø AND B=ØTHENN=6:DB$=E
$(6)
19Ø IF A=Ø AND B=63 THEN N=7:DB$
=E$(7)
2ØØ IF A=Ø AND B=52 THEN N=8:DB$
=E$(8)
21Ø IF A=Ø AND B=4Ø THEN N=9:DB$
=E$(9)
22Ø IF A=Ø AND B=7 THEN N=1Ø:DB$
=E$(1Ø)
23Ø IF N<1 THEN 11Ø
24Ø IF RA=Ø THEN RR=15Ø ELSE RR=
13Ø
25Ø DRAW"C7BM"+STR$(RR)+",11Ø"+D
B$
26Ø PLAY"O4L8CDE"
27Ø IF RA=Ø THEN GU=N:IF GU=1Ø T
HEN GU=Ø
28Ø RA=1:GG=(N*1Ø)+GU
29Ø REM"GG IS THE CHILD'S ANSWER
"
3ØØ NEXT YY
31Ø IF GG=AN THEN PLAY"O3L2ØCDEF
GCDEFG":GOTO 38Ø
32Ø REM"DRAW THE CORRECT ANSWER"
33Ø AN$=STR$(AN)
34Ø A2$=LEFT$(AN$,2):M=VAL(A2$)
35Ø A1$=RIGHT$(AN$,1):N=VAL(A1$)
:IF N=Ø THEN N=1Ø
36Ø DRAW"BM13Ø,16Ø"+E$(M)
37Ø DRAW"BM15Ø,16Ø"+E$(N)
38Ø FOR T=1 TO 3ØØØ:NEXT T
39Ø GOTO 7Ø
4ØØ REM"THE LETTERS AND NUMBERS
NEEDED ARE DRAWN HERE"
41Ø A$="BEHUNU2R4NU2DGL2BGBL6"
42Ø D$="BEHU2ER3D4L3BGBL6"
43Ø I$="BR2BUU4BU2BD7BL8"
44Ø N$="BUU4F4U4BG5BL5"
45Ø O$="BEHU2ER2FD2GL2BGBL6"
46Ø T$="BUR2NU4R2BDBL1Ø"
47Ø E$(1Ø)="BEHU2ER2FD2GL2BGBL6"
48Ø E$(1)="BE2NU3DEBFBGBL9"
49Ø E$(2)="BENR3HER3U2L4BG5BL"
5ØØ E$(3)="BENR3HENR2HER3BG5BL5"
51Ø E$(4)="BENU4E3L4BG4BL2"
52Ø E$(5)="BER4U2L3HER3BG5BL5"
53Ø E$(6)="BU2FR2EU2NHGL2HER2BG5
BL4"
54Ø E$(7)="BUNR4UE3BG5BL4"
55Ø E$(8)="BER2EHEHL2GFNR2GFBGBL
6"
56Ø E$(9)="BER2EHL2GNFU2ER2FBG4B
L6"
57Ø SP$="BE4BUBG5BL5":'*SPACER
58Ø RETURN
59Ø PCLS:PMODE3,1:SCREEN1,1:PCLS
5
6ØØ COLOR6:LINE(2Ø,25)-(235,152)
,PSET,B
61Ø LINE(15,2Ø)-(24Ø,157),PSET,B
62Ø PAINT(17,22),7,6
63Ø DRAW"S12A2BM65,2"+A$+D$+D$+I
$+T$+I$+O$+N$
64Ø Q=RND(4):R=RND(9):S=RND(4):T
=RND(9)
65Ø REM"THE ANSWER"
66Ø AN=(Q*1Ø)+R+(S*1Ø)+T
67Ø DRAW"C6BM13Ø,5Ø"+E$(Q)
68Ø DRAW"BM15Ø,5Ø"+E$(R)
69Ø DRAW"BM13Ø,8Ø"+E$(S)
7ØØ DRAW"BM15Ø,8Ø"+E$(T)
71Ø LINE(85,9Ø)-(95,9Ø),PSET:LIN
E(9Ø,85)-(9Ø,95),PSET
72Ø LINE(85,1ØØ)-(17Ø,1Ø2),PSET,
BF
73Ø RETURN
```

HINT: DON'T BE FOOLED BY THOSE ERROR MESSAGES. Many of our readers have written to us saying the computer tells them there is an error in a line that has been entered correctly. This is usually caused by errors in a related line. For instance, an illegal function call (FC) error is almost always caused by an error elsewhere in the listing. Also, if a line containing a READ statement is typed in wrong, you will get a syntax (SN) error in the DATA statement line number - even though you typed it exactly as it appeared. Remember, be patient and thorough when debugging.

32K
Disk

Trivia Tic-Tac-Toe

By Rick Pitel

One of the problems with quiz or trivia games on the computer is that the answer must be exact. A misspelled word or failure to state the answer exactly the same way the computer sees it gets it wrong. On the other hand, the multiple-choice type question lends itself very well to computer logic. This is the idea behind Tic-Tac-Toe.

The format is that of a tic-tac-toe game with nine categories from which to choose. It uses low resolution graphics and can be run from disk or tape. The program is generally self-explanatory and can be played from the keyboard or with joysticks. If done from the keyboard, first enter the number of the square to be chosen and then one through four to choose the answer. With joysticks, 'X' uses the right joystick and 'O' uses the left joystick. Choose the square by positioning the joystick according to the flashing indicator and push the button. Choose the answer in the same way.

If the answer is correct, an 'X' or 'O' will appear in the square. If the answer is wrong, the correct answer is shown and another category is placed in that square. The game works by pulling a data file holding 100 questions into an array, then feeding multiple-choice questions to the squares as needed.

In order to create the data file it was necessary to write a utility program (Listing 2). The utility program is menu driven and allows for creating and saving the data file on disk or tape. If using the disk, the data file is named to distinguish it from any others you may create. The utility program walks you through the creation of the data file asking for "category", "questions", "correct answer" and three "bogus answers". It then srambles the answers into random order.

Because of the screen format, the category is limited to seven letters, the question is limited to one line and the four answers are limited to two lines. The program will let you know if the entry is too long. I have included a data file that consists of 100 trivia questions.

250	115
540	68
760	137
990	233
1160	112
1400	245
END	136

Listing 1: TICTACTO

```
1ØØ REM  ** TRIVIA TAC TOE **
11Ø REM * RICK PITEL 1984 *
12Ø CLEAR 12ØØØ
13Ø DIM Q$(1ØØ,3):DIM AN(1ØØ):DI
M SQ(9)
14Ø CLS:PRINT:PRINT
141 INPUT"DATA FILE ON 1-CASSETT
E OR       2-DISK";A
15Ø INPUT"NAME OF DATA FILE";F$
151 IF A=2 THEN16Ø
152 INPUT"TURN ON TAPE RECORDER
AND HIT    <ENTER>";A$
153 OPEN"I",-1,F$
154 FOR I=1 TO 1ØØ
155 INPUT#-1,Q$(I,1),Q$(I,2),Q$(
I,3),AN(I)
156 NEXTI
157 CLOSE-1:GOTO2ØØ
16Ø OPEN"I",1,F$
17Ø FOR I=1 TO 1ØØ
18Ø INPUT #1, Q$(I,1),Q$(I,2),Q$
(I,3),AN(I)
19Ø NEXT I
2ØØ FOR I=1 TO 9
21Ø SQ(I)=I:NEXT
22Ø CLOSE 1
23Ø K=Ø
24Ø CLS3:INPUT"CONTROL WITH <1>K
EYBOARD    OR    <2>JOYSTICKS";CC
25Ø IF CC>2 THEN 24Ø
26Ø C$=CHR$(2Ø7)
27Ø BUF$=C$+C$+C$+C$+C$+C$+C$
28Ø ' INITIAL DRAW #
29Ø CLS5
3ØØ FOR L=96 TO 127
31Ø PRINT@L,CHR$(2Ø4);
32Ø PRINT@L+128,CHR$(2Ø4);
```

```
33Ø NEXT L
34Ø FOR L=1Ø TO 362 STEP32
35Ø PRINT@L,CHR$(128);
36Ø PRINT@L+1Ø,CHR$(128);
37Ø NEXT L
38Ø FOR S=1 TO 9
39Ø K=K+1
4ØØ GOSUB 47Ø
41Ø PRINT@BP,S;
42Ø PRINT@BP+31,Q$(K,1);
43Ø NEXT S
44Ø P=1:NF=Ø
45Ø GOTO 58Ø
46Ø ' SUBROUTINE S TO BP
47Ø ON S GOTO 48Ø,49Ø,5ØØ,51Ø,52
Ø,53Ø,54Ø,55Ø,56Ø
48Ø BP=3:RETURN
49Ø BP=13:RETURN
5ØØ BP=23:RETURN
51Ø BP=131:RETURN
52Ø BP=141:RETURN
53Ø BP=151:RETURN
54Ø BP=259:RETURN
55Ø BP=269:RETURN
56Ø BP=279:RETURN
57Ø ' PLAY
58Ø PRINT@384
59Ø IF P=1 THENPRINT@384,"X TURN
";ELSE PRINT@384,"O TURN";
6ØØ IF CC=2 THEN 126Ø
61Ø I$=INKEY$:IF I$=""THEN 61Ø
62Ø S=VAL(I$):IF S=Ø THEN 61Ø
63Ø PRINT@48Ø," ***********  "S"
 *********  ";
64Ø IF S>9 THEN 59Ø
65Ø IF ABS(SQ(S))>1ØØ THEN 59Ø
66Ø GOSUB 47Ø
67Ø PRINT@BP+64,CHR$(191)+CHR$(1
91)+CHR$(191)+CHR$(191)+CHR$(191
);
68Ø PRINT@384,Q$(SQ(S),2)
69Ø PRINT@416,Q$(SQ(S),3)
7ØØ IF CC=2 THEN 145Ø 'JOYSTICK
CONTROL
71Ø I$=INKEY$:IF I$=""THEN 71Ø
72Ø A=VAL(I$):IF A=Ø THEN 71Ø
73Ø PRINT@48Ø,A;
74Ø PRINT@BP+31,BUF$;:PRINT@BP+6
3,BUF$;
75Ø IF A=AN(SQ(S)) THEN 85Ø 'RIG
HT
76Ø PRINT@485,"WRONG-IT WAS"AN(S
Q(S));
77Ø SOUND 1,6:K=K+1
78Ø IF K>1ØØ THEN 118Ø
79Ø IF Q$(K,1)="X" THEN 118Ø
8ØØ PRINT@BP,S;
81Ø PRINT@BP+31,Q$(K,1);
82Ø SQ(S)=K
83Ø IF P=1 THEN P=2 ELSE P=1
84Ø GOTO 58Ø
85Ø PRINT@485,"CORRECT          ";
86Ø SOUND 18Ø,6:NF=NF+1
87Ø A$=CHR$(195):B$=CHR$(2Ø4):C$
=CHR$(2Ø7)
88Ø IF P=2 THEN SQ(S)=-3ØØ:GOTO9
4Ø
89Ø SQ(S)=3ØØ:PRINT@384,"X WINS
THE SQUARE";
9ØØ PRINT@BP,A$+B$+C$+B$+A$;
91Ø PRINT@BP+32,C$+C$+CHR$(128)+
C$+C$;
92Ø PRINT@BP+64,B$+A$+C$+A$+B$;
93Ø GOTO 99Ø
94Ø PRINT@BP,C$+B$+CHR$(128)+B$+
C$;
95Ø PRINT@BP+32,CHR$(128)+C$+C$+
C$+CHR$(128);
96Ø PRINT@BP+64,C$+A$+CHR$(128)+
A$+C$;
97Ø PRINT@384,"O WINS THE SQUARE
98Ø ' CHECK FOR WINNER
99Ø A=SQ(1)+SQ(2)+SQ(3):IF ABS(A
)=9ØØ THEN 111Ø
1ØØØ A=SQ(4)+SQ(5)+SQ(6):IF ABS(
A)=9ØØ THEN 111Ø
1Ø1Ø A=SQ(7)+SQ(8)+SQ(9):IF ABS(
A)=9ØØ THEN 111Ø
1Ø2Ø A=SQ(1)+SQ(4)+SQ(7):IF ABS(
A)=9ØØ THEN 111Ø
1Ø3Ø A=SQ(2)+SQ(5)+SQ(8):IF ABS(
A)=9ØØ THEN 111Ø
1Ø4Ø A=SQ(3)+SQ(6)+SQ(9):IF ABS(
A)=9ØØ THEN 111Ø
1Ø5Ø A=SQ(1)+SQ(5)+SQ(9):IF ABS(
A)=9ØØ THEN 111Ø
1Ø6Ø A=SQ(3)+SQ(5)+SQ(7):IF ABS(
A)=9ØØ THEN 111Ø
1Ø7Ø IF NF=9 THENPRINT@384,"GAME
 OVER":GOTO 113Ø
1Ø8Ø IF P=1 THEN P=2 ELSE P=1
1Ø9Ø GOTO 58Ø
11ØØ ' WINNER
111Ø IF A=9ØØ THEN A$="X"ELSE A$
="O"
112Ø PRINT@384,A$" WINS THE GAME
!"
113Ø PRINT
114Ø IF K>9Ø THENPRINT"INSUFFICI
ENT QUESTIONS LEFT FOR ANOTHER G
AME":STOP
115Ø INPUT"ANOTHER GAME <Y OR N>
";A$
116Ø IF LEFT$(A$,1)="Y" THEN 121
Ø ' REPEAT GAME
117Ø STOP
118Ø PRINT@384,"YOU ARE OUT OF Q
UESTIONS"
119Ø STOP
12ØØ ' REPEAT GAME
```

```
121Ø FOR I=1 TO 9
122Ø SQ(I)=K+I
123Ø NEXT I
124Ø GOTO 29Ø
125Ø REM  JOYSTICK CONTROL
126Ø B$=CHR$(175)+CHR$(175)+CHR$
(175):C$=CHR$(2Ø7)+CHR$(2Ø7)+CHR
$(2Ø7)
127Ø IF P=2 THEN 136Ø
128Ø X=JOYSTK(Ø):Y=JOYSTK(1)
129Ø S=INT(X/22)+1+INT(Y/22)*3
13ØØ IF ABS(SQ(S))>1ØØ THEN 128Ø
131Ø GOSUB47Ø
132Ø PRINT@BP+65,B$;
133Ø B=PEEK(6528Ø)
134Ø IF B=126 OR B=124 OR B=254
OR B=252 THEN 67Ø
135Ø PRINT@BP+65,C$;:GOTO 128Ø
136Ø A=JOYSTK(Ø)
137Ø X=JOYSTK(2):Y=JOYSTK(3)
138Ø S=INT(X/22)+1+INT(Y/22)*3
139Ø IF ABS(SQ(S))>1ØØ THEN 136Ø
14ØØ GOSUB47Ø
141Ø PRINT@BP+65,B$;
142Ø B=PEEK(6528Ø)
143Ø IF B=125 OR B=124 OR B=253
OR B=252 THEN 67Ø
144Ø PRINT@BP+65,C$;:GOTO 136Ø
145Ø B=PEEK(6528Ø):IF B=127 OR B
=255 THEN 146Ø ELSE 145Ø
146Ø IF P=2 THEN 152Ø
147Ø A=INT(JOYSTK(Ø)/16)+1
148Ø PRINT@BP+65,A;
149Ø B=PEEK(6528Ø)
15ØØ IF B=126 OR B=124 OR B=254
OR B=252 THEN 73Ø
151Ø GOTO147Ø
152Ø X=JOYSTK(Ø):A=INT(JOYSTK(2)
/16)+1
153Ø PRINT@BP+65,A;
154Ø B=PEEK(6528Ø)
155Ø IF B=125 OR B=124 OR B=253
OR B=252 THEN 73Ø
156Ø GOTO152Ø
```

180	165
406	229
530	105
700	75
850	126
1110	147
1250	4
1540	193
END	15

Listing 2: TRIVUTIL

```
1Ø REM **UTILITY PROGRAM**
15 CLEAR 12ØØØ
2Ø DIM Q$(1ØØ,3):DIM AN(1ØØ):DIM
 R$(4)
5Ø A$="XXXXXXX"
6Ø B$="XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXX"
1ØØ CLS
11Ø PRINT"THIS IS THE PROGRAM TH
AT CREATES & MODIFIES THE QUESTI
ON/ANSWER
12Ø PRINT"FILES FOR THE TRIVIA-T
AC-TOE    GAME"
125 PRINT"PICK ONE OF THE FOLLOW
ING"
13Ø PRINT"1-CREATE A NEW QUESTIO
N/ANSWER  FILE"
135 PRINT"2-MODIFY OR CORRECT AN
 EXISTING QUESTION/ANSWER FILE"
14Ø PRINT"3-ADD TO A FILE THAT I
S NOT FULL YET<LESS THAN 1ØØ QUE
STIONS>"
15Ø PRINT"4-LOADING AND UNLOADIN
G TO DISK OR TAPE<MOVING THE DAT
A FILE>"
17Ø FOR I=1 TO 1ØØ
18Ø Q$(I,1)=A$:Q$(I,2)=B$:Q$(I,3
)=B$+B$:AN(I)=Ø
19Ø NEXT I
2ØØ INPUT MC
21Ø IF MC=1 THEN 5ØØ
22Ø IF MC<4 THEN 29Ø
23Ø CLS
26Ø PRINT"1-LOAD FILE"
27Ø PRINT"2-SAVE FILE"
28Ø INPUT A:IF A=2 THEN 41Ø
29Ø GOSUB 34Ø
3ØØ ON MC GOTO 31Ø,13ØØ,14ØØ,15Ø
Ø
31Ø PRINT"ERROR":GOTO 1ØØ
34Ø ' LOAD FILE SUBROUTINE
345 INPUT"1-TAPE OR 2-DISK";A
347 IF A=2 THEN 39Ø
35Ø INPUT"TURN TAPE RECORDER ON
& HIT     <ENTER>";A$
355 OPEN"I",-1,"TTDATA"
36Ø FOR I=1 TO 1ØØ
365 INPUT#-1,Q$(I,1),Q$(I,2),Q$(
I,3),AN(I)
37Ø NEXT I
375 CLOSE-1
38Ø RETURN
39Ø INPUT"ENTER FILE NAME";F$
4Ø1 OPEN "I",1,F$
4Ø2 FOR I=1TO1ØØ
4Ø3 INPUT #1,Q$(I,1),Q$(I,2),Q$(
I,3),AN(I)
4Ø4 NEXT I
4Ø5 CLOSE 1
4Ø6 RETURN
41Ø ' SAVING THE FILE
415 INPUT"1-TAPE OR 2-DISK";A
42Ø IF A=2 THEN 46Ø
425 INPUT"TURN TAPE RECORDER ON
RECORD &  HIT <ENTER>";A$
```

```
43Ø OPEN"O",-1,"TTDATA"
435 FOR I=1 TO 1ØØ
44Ø PRINT#-1,Q$(I,1),Q$(I,2),Q$(
I,3),AN(I)
445 NEXT I
45Ø CLOSE-1
455 GOTO 47Ø
46Ø INPUT"ENTER FILE NAME";F$
464 OPEN "O",1,F$
465 FOR I=1TO1ØØ
466 WRITE #1,Q$(I,1),Q$(I,2),Q$(
I,3),AN(I)
467 NEXT I
468 CLOSE 1
47Ø PRINT"DATA LOADED"
475 INPUT"1-CONTINUE 2-QUIT";A
48Ø IF A=1 THEN 15ØØ
49Ø STOP
5ØØ CLS
51Ø PRINT"OK, HERE'S THE DEAL"
52Ø PRINT"THIS NEW FILE CAN TAKE
 1ØØ MULTIPLE CHOICE QUESTIONS-E
ACH HAS 4 PARTS"
53Ø PRINT"1-CATEGORY OF QUESTION
-SUCH AS  HISTORY,PEOPLE,TV,SPOR
TS,COMICS"
535 PRINT"---MUST 7 OR LESS LETT
ERS"
54Ø PRINT"2-THE QUESTION--KEEP T
O ONE LINE  <32 LETTERS MAX>"
55Ø PRINT"3-THE CORRECT ANSWER"
56Ø PRINT"4-THREE BOGUS ANSWERS"
57Ø PRINT"THE COMPUTER WILL PUT
THESE ANSWERS IN RANDOM ORDER"
572 INPUT"HIT <ENTER> TO CONTINU
E";A$
575 PRINT"LENGTH OF THE 4 ANSWER
S MUST BE LESS THAN 2 LINES LONG
"
58Ø PRINT"BECAUSE OF THE WAY OUR
 COMPUTER TAKES DATA--DON'T USE
ANY COMMAS"
585 PRINT"IF YOU WANT TO QUIT EA
RLY, TYPE IN STOP WHEN IT ASKS F
OR CATEGORY"
59Ø PRINT"IF YOU DON'T LIKE THE
WAY THE   LAST QUESTION TURNED O
UT TYPE IN REDO"
6ØØ A=TIMER
61Ø IF A>5ØØ THEN A=A-5ØØ:GOTO61
Ø
62Ø FOR I=1 TO A:B=RND(A):NEXT
63Ø INPUT"PRESS <ENTER> TO START
";A$
7ØØ K=1:CLS5
71Ø PRINT"QUESTION"K
72Ø INPUT"CATEGORY";A$
73Ø IF A$="STOP"THEN 15ØØ
74Ø IF A$="REDO"THEN K=K-1:GOSUB
11ØØ:GOTO95Ø
75Ø IF LEN(A$)>7 THENPRINT"7 LET
TERS MAX PLEASE REDO":GOTO72Ø
76Ø Q$(K,1)=A$
77Ø PRINT"ENTER THE QUESTION":IN
PUT A$
78Ø IF LEN(A$)>32 THENPRINT"THIS
 CAN BE ONLY 1 LINE, ACE":GOTO77
Ø
79Ø Q$(K,2)=A$
8ØØ PRINT"ENTER THE CORRECT ANSW
ER":INPUT A$
81Ø PRINT"ENTER FIRST BOGUS ANSW
ER":INPUT B$
82Ø PRINT"ENTER ANOTHER BOGUS AN
SWER":INPUT C$
83Ø PRINT"ENTER ONE MORE BOGUS A
NSWER":INPUT D$
84Ø B=LEN(A$+B$+C$+D$)
845 IF B>52 THENPRINT"YOUR ANSWE
R IS"52-B"LETTERS TOO LONG-TRY A
GAIN":GOTO8ØØ
85Ø GOSUB 1ØØØ
86Ø IF L<27 AND LL<27 THEN 92Ø
87Ø GOSUB 1ØØØ
88Ø IF L<27 AND LL<27 THEN 92Ø
89Ø GOSUB 1ØØØ
895 IF L<27 AND LL<27 THEN 92Ø
9ØØ Q$(K,3)="1-"+R$(1)+" 2-"+R$(
2)+" 3-"+R$(3)+" 4-"+R$(4)
91Ø GOTO 95Ø
92Ø D=27-L:A$="1-"+R$(1)+" 2-"+R
$(2)
93Ø FOR I=1 TO D:A$=A$+" ":NEXT
94Ø Q$(K,3)=A$+"3-"+R$(3)+" 4-"+
R$(4)
95Ø CLS RND(8)
96Ø PRINT Q$(K,1),AN(K)
97Ø PRINT Q$(K,2):PRINT Q$(K,3)
98Ø K=K+1:IF K>1ØØ THENPRINT"FIL
E FULL!":GOTO 17ØØ
99Ø GOTO 71Ø
999 ' RANDOM ANSWER SUBROUTINE
1ØØØ A=RND(4):AN(K)=A
1Ø1Ø B=RND(3):IF B=A THEN B=B+1
1Ø2Ø C=RND(2):IF C=B OR C=A THEN
 C=C+1
1Ø3Ø IF C=B OR C=A THEN C=C+1
1Ø4Ø D=1Ø-A-B-C
1Ø5Ø R$(A)=A$:R$(B)=B$:R$(C)=C$:
R$(D)=D$
1Ø6Ø L=LEN(R$(1)+R$(2)):LL=LEN(R
$(3)+R$(4))
1Ø7Ø RETURN
11ØØ CLS
111Ø PRINT Q$(K,1):PRINT Q$(K,2)
:PRINT Q$(K,3)
112Ø PRINT"DO YOU WANT TO CORREC
T"
1121 PRINT"1-CATEGORY  2-QUESTIO
```

```
N"
1130 PRINT"3-ANSWER      4-THE WHO
LE THING";
1140 INPUT A:ON A GOTO 1160,1190
,1230,1160
1150 RETURN
1160 INPUT"CATEGORY";A$
1170 IF LEN(A$)>7 THENPRINT"7 LE
TTERS OR LESS PLEASE":GOTO 1160
1180 Q$(K,1)=A$:IF A=1 THEN RETU
RN
1190 PRINT"QUESTION":INPUT Q$(K,
2)
1210 IF LEN(Q$(K,2))>32 THENPRIN
T"THIS ISN'T A NOVEL-1 LINE PLEA
SE":GOTO1190
1215 IF A=4 THEN 1230
1220 RETURN
1230 PRINT"ENTER ALL MULTIPLE CH
OICE          ANSWERS & NUMBERS":IN
PUT A$
1240 IF LEN(A$)>64 THENPRINT"THA
T'S TOO LONG-REDUCE TO 2 LINES":
GOTO1230
1250 IF A=3 THEN 1270
1270 Q$(K,3)=A$
1280 INPUT"WHICH IS THE CORRECT
ANSWER";B
1290 IF B<1 THEN 1280
1295 AN(K)=B:RETURN
1299 'MODIFY OR CORRECT
1300 CLS:PRINT"ENTER 999 TO SCRO
LL THRU ALL      QUESTIONS"
1310 PRINT"ENTER QUESTION # TO M
ODIFY          0 TO STOP";
1320 INPUT K
1330 IF K=999 THEN 1600
1340 IF K=0 THEN 1500
1350 GOSUB 1100
1360 GOTO 1310
1399 ' ADD TO EXISTING FILE
1400 PRINT"LOAD EXISTING FILE"
1420 K=1
1430 IF Q$(K,1)="XXXXXXX" THEN 1
460
1440 K=K+1:IF K<=100 THEN 1430
1450 PRINT"FILE IS FULL":GOTO170
0
1460 CLS5
1470 GOTO 710
1499 ' SECONDARY MENU
1500 CLS 5
1510 PRINT"1-MODIFY OR CORRECT"
1520 PRINT"2-ADD TO THIS FILE"
1530 PRINT"3-SCROLL THRU ALL QUE
STIONS"
1540 PRINT"4-TERMINATE PROGRAM"
1550 INPUT MS
1560 ON MS GOTO 1310,1420,1600,1
700
1570 GOTO 1500
1599 '   SCROLL
1600 PRINT"THIS OPTION WILL PAGE
 THRU ALL  QUESTIONS-4 AT A TIME
"
1610 PRINT"ENTER S TO STOP-ANY O
THER KEY TO CONTINUE"
1620 INPUT"HIT <ENTER> TO BEGIN
PAGING";A$
1625 CLS
1630 FOR I=1 TO 100
1640 PRINT I,Q$(I,1)
1650 PRINT Q$(I,2):PRINT Q$(I,3)
1660 IF INT(I/4)<>I/4 THEN 1690
1670 I$=INKEY$
1680 IF I$="" THEN 1670
1685 IF I$="S" THEN 1500
1690 NEXT I
1695 GOTO 1500
1699 ' TERMINATE
1700 INPUT"1-SAVE FILE      2-JUS
T STOP";A
1710 IF A=2 THEN STOP
1740 GOTO 410
1750 END
```

Editor's Note: The text that follows is the sample data file for this program. You can type it in using a word processor, and save it on disk in ASCII, or you can use it as a reference when running Listing 2 to create a file.

Listing 3: TRIVDATA

```
 "PEOPLE","WHO WAS THE ELEPHANT MAN","1-
JOHN MERRICK 2-JIM MORRISON   3-BEN HOAR
Y 4-ORSON WELLS", 1
"HISTORY","LED RAID ON HARPERS FERRY 185
9","1-VALERIE HARPER 2-GEN SHERMAN 3-JOH
N BROWN 4-STONEWALL JACKSON", 3
"WORLD","IDI AMIN WAS DICTATOR OF","1-SW
EDEN 2-UGANDA                3-ANGOLA 4-L
IBYA", 2
"TV","MURRAY THE COP IS FROM","1-BARNEY
MILLER 2-MAYBERRY RFD  3-ODD COUPLE 4-AR
CHIE BUNKER", 3
"USA","STATUE OF LIBERTY IS MADE OF","1-
WOOD 2-COPPER                3-STONE 4-
MARSHMELLOWS", 2
"BOOKS","BILBO BAGGINS COUSIN","1-DONALD
 2-GOLLUM                3-FRODO 4-ROLLO"
, 3
"MUSIC","BRIAN WILSONS GROUP","1-BEACH B
OYS 2-BEEGEES            3-DOORS 4-LED ZEP
PELIN", 1
"SPORTS","PANCHO GONZALES PLAYS","1-TENN
IS 2-BASEBALL                3-SOCCER 4-GOL
F", 1
"INVENT","INVENTED THE ELECTRIC RAZOR","
1-K GILLETTE 2-JACOB SCHICK      3-WILSON
```

```
 YOUNG 4-GABBY HAYES", 2
"ANIMALS","WHICH IS NOT A REPTILE","1-SA
LAMANDER 2-IGUANA              3-CHAMELEON
4-RATTLESNAKE", 1
"CARS","WHO MAKES THE GLC","1-MAZDA 2-VO
LVO                3-AMC 4-GMC", 1
"GEOG","WHERE IS THE BLACK FOREST","1-GE
RMANY 2-USSR                  3-FINLAND 4-
ANGOLA", 1
"MOVIES","STAR OF BECKET","1-PETER OTOOL
E 2-REX HARRISON   3-ED ASNER 4-R BURTON
", 4
"CAPITAL","CAPITAL OF AUSTRALIA","1-CANB
ERRA 2-AUCKLAND            3-SYDNEY 4-PER
TH", 1
"LEGENDS","SON OF SIR LANCELOT","1-JUNIO
R 2-ULYSSES                3-GUY 4-GALAHAD
", 4
"PRES","JACKIE KENNEDYS MAIDEN NAME","1-
BOUVIER 2-STAPLES              3-ONASSIS
4-FENWICK", 1
"SPOUSES","FORMER WIFE OF CARY GRANT","1
-DYAN CANNON 2-ANGIE DICKENSON 3-JEAN SI
MMONS 4-EVA GABOR", 1
"COMICS","BROOM HILDAS FRIEND","1-MAYNAR
D 2-FRED                   3-GAYLORD 4-CASP
ER", 3
"MEDICAL","DISEASE OF GUMS","1-PHLEBITIS
 2-DOUBLEMINT        3-PLEURISY 4-PYORRH
EA", 4
"DISNEY","VOICE OF DONALD DUCK","1-WALTE
R LANTZ 2-WALT DISNEY    3-CLARENCE NASH
 4-MEL BLANC", 3
"PEOPLE","KNOWN AS GENTLEMAN JIM","1-JIM
MY SNYDER 2-JIM PRICE       3-JIM NAYBORS
 4-JAMES CORBETT", 4
"HISTORY","DEFEATED NAPOLEON AT WATERLOO
","1-HENRY 2-WELLINGTON             3-EDW
ARD 4-HANNIBAL", 2
"WORLD","WHERE IS COPACABANA BEACH","1-R
IO DE JANEIRO 2-HAWAII         3-CUBA 4-HO
LLYWOOD", 1
"TV","ON WHAT SHOW IS PIGS IN SPACE","1-
SOLID GOLD 2-GREEN ACRES        3-STAR TRE
K 4-MUPPETS", 4
"USA","DISCOVERED THE COLUMBIA RIVER","1
-CAPT ROBT GRAY 2-WM COLUMBIA  3-LEWIS &
 CLARK 4-HARRY TRUMAN", 1
"BOOKS","AUTHOR OF BRAVE NEW WORLD","1-H
.G.WELLS 2-GERALD BROWN        3-ALDOUS HU
XLEY 4-LAURA HOBSON", 3
"MUSIC","KAREN CARPENTERS BROTHER","1-WI
LL 2-BOB                      3-CLAUDE 4-R
ICHARD", 4
"SPORTS","LOCATION OF FENWAY PARK","1-CH
ICAGO 2-BALTIMORE             3-SAN DIEGO
4-BOSTON", 4
"SCIENCE","OSTEOLOGY IS THE STUDY OF","1
-BLENDERS 2-FISH                 3-BONES 4
-BIRDS", 3
```

```
"ANIMALS","LARGEST LAND CARNIVORE","1-JO
HN BELUSHI 2-KODIAK BEAR    3-RHINOCEROU
S 4-GRIZZLY BEAR", 2
"CARS","WHO MAKES THE SILVER SPIRIT","1-
SUBARU 2-FORD                  3-ROLLS RO
YCE 4-TOYOTA", 3
"GEOG","WHERE IS BOTSWANA","1-MIDDLE EAR
TH 2-SOUTH AFRICA    3-ASIA 4-ORIENT", 2
"MOVIES","WHO PLAYED THE WIZARD OF OZ","
1-DICK CLARK 2-FRANK MORGAN      3-RAY BO
LGER 4-JACK HALEY", 2
"CAPITAL","CAPITAL OF WISCONSIN","1-MILW
AUKEE 2-ROCKFORD            3-MADISON 4-CO
LUMBUS", 3
"LEGENDS","GREEK KING WHO FOUGHT TROJANS
","1-AGAMEMNON 2-ULYSSES             3-ALE
XANDER 4-APOLLO", 1
"PRES","KNOWN AS OLD ROUGH & READY","1-H
ERBERT HOOVER 2-ZACH TAYLOR  3-F.D.ROOSE
VELT 4-GERALD FORD", 2
"SPOUSES","FORMER HUSBAND OF JULIE LONDO
N","1-JACK WEBB 2-JOHN WARNER         3-XA
VIER CUGAT 4-WOODY ALLEN", 1
"COMICS","DAGWOODS BOSS","1-LOU GRANT 2-
MR BEASLEY          3-MR DITHERS 4-MR BIGD
OME", 3
"SPACE","MAKEUP OF THE SUN","1-HYDROGEN
2-OXYGEN              3-AMMONIA 4-URANIUM
", 1
"DISNEY","DUMBOS MOUSE FRIEND","1-THUMPE
R 2-SPEEDY                3-JERRY 4-TIMOTH
Y", 4
"PEOPLE","WHO WAS KNOWN AS SCARFACE","1-
MUSSOLINI 2-BUGSY MALONE        3-AL CAPON
E 4-QUASIMOTO", 3
"HISTORY","TRIED TO BLOW UP HOUSE OF LOR
DS","1-RICHARD STARKY 2-GUY FAWKES    3-O
LIVER CROMWELL 4-JOHN CALVIN", 2
"WORLD","ANWAR SADAT'S COUNTRY","1-TURKE
Y 2-IRAQ                  3-EGYPT 4-ISREA
L", 3
"TV","D.A. IN PERRY MASON","1-PERRY WHIT
E 2-IRONSIDE          3-HAMILTON BURGER 4-
PAUL DRAKE", 3
"USA","TERM FOR A U.S. SENATOR","1-4 YRS
. 2-2 YRS.                 3-6 YRS. 4-LIFE
", 3
"BOOKS","OCCUPATION OF BOB CRATCHET","1-
BANK TELLER 2-SCHOOL TEACHER  3-FARMER 4
-CLERK", 4
"MUSIC","SONG BY THE COWSILLS","1-SUZY Q
  2-HAIR  3-DIZZY          4-GOIN UP TO THE
 COUNTRY", 2
"SPORTS","DIXIE WALKER PLAYS","1-GOLF 2-
TENNIS                  3-BASEBALL 4-BASK
ETBALL", 3
"INVENT","INVENTOR OF THE CLOCK","1-GEOR
GE WALTHAM 2-THE CHINESE  3-BEN FRANKLIN
 4-WILL SMITH", 2
"ANIMALS","WHAT IS AN EMU","1-REPTILE 2-
```

```
ANTELOPE           3-LARGE BIRD 4-INSEC
T", 3
"CARS","WHO MAKES THE MONDIAL","1-PEUGEO
T 2-FERRARI           3-CHEVY 4-CADILL
AC", 2
"GEOG","SHARES ISLAND WITH HAITI","1-JAM
AICA 2-NEW GUINEA          3-DOMINICAN R
EPUBLIC 4-BORNEO", 3
"MOVIES","DIRECTOR OF STAR TREK III","1-
GENE RODDENBERRY 2-GEO LUCAS  3-LEONARD
NEMOY 4-ARTHUR HILLER", 3
"CAPITAL","CAPITAL OF PORTUGAL","1-MADRI
D 2-PORTO               3-LISBON 4-GIBR
ALTER", 3
"LEGENDS","FLEW TOO CLOSE TO THE SUN","1
-PINOCCHIO 2-ICARUS           3-PROMETH
EUS 4-JOHN GLENN", 2
"PRES","WHO SHOT PRESIDENT REAGEN","1-MA
RK CHAPMAN 2-JAMES EARL RAY 3-SIRHAN SIR
HAN 4-JOHN HINKLEY", 4
"SPOUSES","FORMER HUSBAND OF URSULA ANDR
ES","1-JOHN HUSTON 2-DEAN MARTIN     3-D
ICK POWELL 4-JOHN DEREK", 4
"COMICS","BATMAN'S TRUE IDENTITY","1-LAM
ONT CRANSTON 2-BRUCE WAYNE 3-DICK GRAYSO
N 4-CLARK KENT", 2
"MEDICAL","ADENOIDS ARE LOCATED IN THE",
"1-FOOT 2-THROAT                3-INTES
TINES 4-LIVER", 2
"DISNEY","WALT DISNEY'S MIDDLE NAME","1-
ELIAS 2-GORDON                 3-FLETCHER
 4-DONALD", 1
"PEOPLE","HUNTER OF FUGITIVE NAZIS"."1-S
TEVE MCQUEEN 2-ALBERT SPEER  3-SIMON WIE
SENTHAL 4-A EICHMANN", 3
"HISTORY","CONFEDERATE PRESIDENT","1-ROB
T E LEE 2-JEFFERSON DAVIS  3-WM SHERMAN
4-BOSS HOGG", 2
"WORLD","BELIZE IS A COLONY OF","1-ENGLA
ND 2-FRANCE              3-USA 4-CHILE",
 1
"TV","HOME OF OLIVER WENDELL DOUGLAS","1
-MAYBERRY 2-HONG KONG          3-WALNUT
GROVE 4-HOOTERVILLE", 4
"USA","FIRST NATIONAL PARK","1-SMOKEY MO
UNTAINS 2-YOSEMITE   3-YELLOWSTONE 4-ZIO
N", 3
"BOOKS","WHO WROTE 'SAVING THE QUEEN'","
1-ROBT GREEN 2-SIDNEY SHELTON   3-WM F B
UCKLEY 4-DAVID BOWIE", 3
"MUSIC","WHO WROTE 'CLAIR DE LUNE'","1-E
LTON JOHN 2-TOULOUSE LAUTREC 3-CHICK COR
EA 4-CLAUDE DEBUSSY", 4
"SPORTS","CHICAGO HOCKEY TEAM","1-RED SO
X 2-BLACK HAWKS         3-BEARS 4-BRUINS
", 2
"INVENT","EDISON DID NOT INVENT THE","1-
ELECTRIC LAMP 2-PHONOGRAPH    3-RADIO 4-
MOVIE CAMERA", 3
"ANIMALS","A COATIMUNDI IS RELATED TO TH
E","1-RACCOON  2-KIWI               3-PY
THON  4-OAK TREE", 1
"CARS","WHO MAKES THE SKYHAWK","1-ASTON
MARTIN 2-DODGE          3-PLYMOUTH 4-BUI
CK", 4
"GEOG","WHERE IS THE TASMAN SEA","1-THE
MOON 2-WEST AFRICA          3-INDIA 4-NEW
ZEALAND", 4
"MOVIES","STAR OF 'SPARTACUS'","1-WM CON
RAD 2-REX HARRISON 3-KIRK DOUGLAS 4-CHAR
LTON HESTON", 3
"CAPITAL","CAPITAL OF NEW HAMPSHIRE","1-
CONCORD 2-WEBSTER               3-MONTPELI
ER 4-BOSTON", 1
"LEGENDS","HOME OF ULYSSES","1-ITHACA 2-
CAMELOT               3-ROME 4-ATHENS", 1
"PRESDNT","G WASHINGTON'S POLITICAL PART
Y","1-FEDERALIST 2-WHIG              3-DE
MOCRATIC 4-REPUBLICAN", 1
"SPOUSES","FORMER WIFE OF GEORGE PEPPARD
","1-DINAH SHORE 2-ELIZ ASHLEY     3-HOP
E LANGE 4-AVA GARDNER", 2
"COMICS","POGO'S TURTLE FREIND","1-POULE
TT 2-CHURCHY            3-CHARLIE 4-MYR
TLE", 2
"SPACE","FIRST DOG IN SPACE","1-SNOOPY 2
-LAIKA              3-BEN 4-PLUTO", 2
"DISNEY","PINOCCHIO'S CONSCIENCE","1-JIM
INY CRICKET 2-TINKERBELL   3-GEPPETTO 4-
WOOD FAIREY", 1
"PEOPLE","BIRTHPLACE OF DAVID NIVEN","1-
CANADA 2-WALES                 3-SCOTLAND
 4-NEW YORK", 3
"HISTORY","BABYLONIAN KING","1-CYRUS 2-N
EBUCHADNEZZER          3-DARIUS 4-PTOLMY",
 2
"WORLD","PATRON SAINT OF SCOTLAND","1-ST
 CHRISTOPHER 2-ST ANDREW    3-ST LUKE 4-
ST BAGPIPE", 2
"TV","MCLEAN STEVENSON PLAYED","1-MARTIN
 2-SHERMAN POTTER        3-ADAM CARTRIGHT
 4-HENRY BLAKE", 4
"USA","FIRST WOMAN IN CONGRESS","1-JEANE
TTE RANKIN 2-SUSAN ANTHONY 3-SHIRLEY TEM
PLE 4-NEL ROSS", 1
"BOOKS","EGG HATCHING ELEPHANT","1-HORTO
N 2-DUMBO               3-GROUCHO 4-SID
NEY", 1
"MUSIC","FOGERTY BROTHERS PLAYED WITH","
1-DOORS 2-ALLMAN BROS             3-CREEDE
NCE 4-EAGLES", 3
"SPORTS","CY YOUNG AWARD IS FOR","1-PITC
HING 2-FIELDING              3-KICKING 4-TE
NNIS", 1
"SCIENCE","ICHTHYOLOGY IS THE STUDY OF",
"1-REPTILES 2-DUNG                 3-BUGS
4-FISH", 4
"ANIMALS","WHICH IS A CARNIVORE","1-ORYX
 2-MONGOOSE                  3-WALLABY 4-YA
K", 2
```

```
"CARS","WHO MAKES THE BRAT","1-SUBARU 2-
NISSAN              3-JEEP 4-ALPHA ROME
O", 1
"GEOG","1ST COUNTRY SOUTH OF DETROIT","1
-HAITI 2-BRAZIL                3-CANADA
4-MEXICO", 3
"MOVIES","STAR OF DUAL IN THE SUN","1-GR
EG. PECK 2-DENNIS WEAVER    3-JIMMY STUA
RT 4-FRANKIE AVALON", 1
"CAPITAL","CAPITAL OF ITALY","1-FLORENCE
 2-NAPLES             3-ROME 4-MILAN", 3
"LEGENDS","ROMAN EQUIVALENT OF ZEUS","1-
JUPITER 2-NEPTUNE              3-ODIN 4-A
POLLO", 1
"PRESDNT" "PRESIDENT IN 19ØØ","1-HOOVER
2-FDR                    3-MCKINLEY 4-GRAN
T", 3
"SPOUSES","FORMER WIFE OF DICK POWELL","
1-JOAN BLONDELL 2-JUDY CARNE    3-JANE F
ONDA 4-ABBE LANE", 1
"COMICS","THE MOLE & PRUNEFACE WERE IN",
"1-DICK TRACY 2-SPIDERMAN          3-STEVE
 CANYON 4-BATMAN", 1
"MEDICAL","WEAKENING OF ARTERY WALL","1-
ACNE 2-ANEURYSM                 3-APOPLEXY
 4-AUTISM", 2
"DISNEY","ABE LINCOLNS DISNEYLAND ADDRES
S","1-ADVENTURELAND 2-FRONTIERLAND  3-MA
IN ST 4-NEW ORLEANS SQ", 3
```

GAME

32K ECB

The Word-Hunt Crossword Puzzle

by Dennis Teague

Puzzle is an educational game requiring 32K and Extended Colour Basic. It may be reduced to a 16K non-Extended Colour Basic program with a few minor changes. These changes are listed in Table 1 and Table 2.

Puzzle hides up to nine words with a maximum of nine letters in length that you enter on a 10 by 16 grid. The program then asks you to find the words.

To do this, use the arrow keys to move the cursor to the first letter in the word, then press ENTER and the direction of the word (right-arrow key or down-arrow key only). Next, type the number of letters in the word. If the computer finds a match, it prints the word on the grid in inverse video. When all words are found, the computer tells how long you took and asks if you wish to run the program again.

Feel free to modify any part of this program.

Table 1
For 16K Extended Color BASIC
Delete lines 5 through 55
Delete Line 75
Delete lines 685 through 850
Delete Line 870

Table 2
For 16K Non-Extended Color BASIC
Change Line 180 to:
`180 PRINT " ";B$(A);`
Delete Line 245
Delete Line 585
Delete lines 615 through 625, 635, 650
Delete lines 855 through 935
Also, the same as 16K ECB
Also, change all LINE INPUTs to INPUTs

Program Outline

Lines	Function
5-55	Remarks for names
60-95	Get program set up
100-135	Enter words
140	Send to fill routine
145-200	Put board on screen
205-280	Find out what key is pressed and what to do
285-305	Find out direction
310-360	Find out number of letters and create check word
365-435	If check word is the same as a word entered, then change screen to inverse video
440-540	Routine to place word into array A$
545-590	Routine to fill space in array A$
595-610	Routine to tell where cursor is located

615-625	Routine to keep track of how long on this puzzle
630-855	Instructions and title page
860-965	End routine

Variable List

Variables	Function
A$	Main array, also dummy variable
B$	Word array
D$	Test variable
I$	INKEY$ variable
A,B	FOR/NEXT variable
C	Counter
D	Direction of word
H	Hours on puzzle
M	Minutes on puzzle
N	Number of letters
O	Old position
P	New position
Q	Random number
R	Position for word
S	Position for word, also seconds on puzzle
U	Helps figure PRINT@ location for inverse video
X	Number of row
Y	Number of column

140	109	740	180
275	124	790	54
400	158	875	108
530	166	END	82
650	54		

The listing: PUZZLE

```
5 '   *************************
1Ø ' ***     PROGRAM BY:     ***
15 ' ***    DENNIS TEAGUE    ***
2Ø ' ***  455 THURSTON DR.   ***
25 ' ***  NOBLESVILLE, IN    ***
3Ø ' ***        46Ø6Ø        ***
35 ' *************************
4Ø ' ***DOCUMENTATION  HELP***
45 ' ***         BY:         ***
5Ø ' ***   KEITH  REYNOLDS   ***
55 ' *************************
6Ø CLEAR 12ØØ
65 GOSUB 63Ø
7Ø DIM A$(16,1Ø)
75 PRINT"PUZZLE";TAB(26);"PAGE 7
"
8Ø FOR A=1 TO 16
85 FOR B=1 TO 1Ø
9Ø A$(A,B)=CHR$(255)
95 NEXT B,A
1ØØ FOR A=1 TO 9
1Ø5 PRINT"ENTER WORD #";A;
11Ø INPUT B$(A)
115 IF A=1 AND B$(1)="" THEN 1Ø5
12Ø IF B$(A)="" THEN 14Ø
125 IF LEN(B$(A))>9 THEN 1Ø5
13Ø GOSUB 44Ø
135 NEXT A
14Ø GOSUB 545
145 PRINT:PRINT" ";
15Ø FOR A=1 TO 1Ø
155 FOR B=1 TO 16
16Ø PRINTA$(B,A);" ";
165 NEXT B,A
17Ø PRINT
175 FOR A=1 TO 9
18Ø PRINT USING " %        %";B$(
A);
185 C=C+1:IF C=3 THEN PRINT:C=Ø
19Ø NEXT A
195 O=33
2ØØ PRINT@48Ø,"position cursor o
n first letter";
2Ø5 I$=INKEY$
21Ø IF I$=CHR$(8) THEN P=P-2
215 IF I$=CHR$(9) THEN P=P+2
22Ø IF I$=CHR$(1Ø) THEN P=P+32
225 IF I$=CHR$(94) THEN P=P-32
23Ø IF I$=CHR$(13) THEN 285
235 IF P>351 THEN P=351
24Ø IF P<33 THEN P=33
245 GOSUB 615
25Ø IF P=O THEN 265
255 GOSUB 595
26Ø PRINT@O,A$(Y,X);
265 PRINT@P,CHR$(192);:FOR A=1 T
O 5Ø:NEXT A
27Ø PRINT@P,CHR$(255);:FOR A=1 T
O 5Ø:NEXT A
275 O=P
28Ø GOTO 2Ø5
285 PRINT@48Ø,"press arrow showi
ng direction  ";
29Ø I$=INKEY$
295 IF I$=CHR$(9) THEN D=1
3ØØ IF I$=CHR$(1Ø) THEN D=2
3Ø5 IF D=Ø THEN 29Ø
31Ø PRINT@48Ø,"press number of l
etters in word";
315 I$=INKEY$
32Ø N=VAL(I$)
325 IF N=Ø THEN 315
33Ø GOSUB 595
335 IF D=1 AND Y+N-1>16 THEN 38Ø
34Ø IF D=2 AND X+N-1>1Ø THEN 38Ø
345 FOR A=Ø TO N-1
35Ø IF D=1 THEN D$=D$+A$(Y+A,X)
355 IF D=2 THEN D$=D$+A$(Y,X+A)
36Ø NEXT A
365 FOR B=1 TO 9
37Ø IF D$=B$(B) THEN B$(B)="":GO
TO 39Ø
375 NEXT B
38Ø D=Ø:D$=""
385 GOTO 2ØØ
39Ø GOSUB 595
395 FOR A=Ø TO N-1
```

```
400 IF D=1 THEN A$(Y+A,X)=CHR$(A
SC(A$(Y+A,X))+32):PRINT@P+U,A$(Y
+A,X);:U=U+2
405 IF D=2 THEN A$(Y,X+A)=CHR$(A
SC(A$(Y,X+A))+32):PRINT@P+U,A$(Y
,X+A);:U=U+32
410 NEXT A
415 U=0:D=0:D$=""
420 FOR B=1 TO 9
425 IF B$(B)<>"" THEN 200
430 NEXT B
435 GOTO 860
440 'PUT THE WORD SOMEPLACE
445 Q=RND(2)
450 IF Q=2 THEN 500
455 R=RND(16-LEN(B$(A)))
460 S=RND(10)
465 FOR B=R TO R+LEN(B$(A))
470 IF A$(B,S)<>CHR$(255) THEN 4
45
475 NEXT B
480 FOR B=R TO R+LEN(B$(A))-1
485 A$(B,S)=MID$(B$(A),B-R+1,1)
490 NEXT B
495 RETURN
500 R=RND(10-LEN(B$(A)))
505 S=RND(16)
510 FOR B=R TO R+LEN(B$(A))
515 IF A$(S,B)<>CHR$(255) THEN 4
45
520 NEXT B
525 FOR B=R TO R+LEN(B$(A))-1
530 A$(S,B)=MID$(B$(A),B-R+1,1)
535 NEXT B
540 RETURN
545 'FILL REMAINING SPACES
550 PRINT:PRINT"PLEASE WAIT..."
555 FOR A=1 TO 16
560 FOR B=1 TO 10
565 IF A$(A,B)<>CHR$(255) THEN 5
75
570 A$(A,B)=CHR$(RND(26)+64)
575 NEXT B,A
580 CLS
585 S=0
590 RETURN
595 'FORUMLA FOR POSITION
600 X=INT(O/32)
605 Y=((O+1)-(32*X))/2
610 RETURN
615 'TIMER SECTION
620 IF TIMER>61 THEN S=S+1:TIMER
=0
625 RETURN
630 CLS
635 PRINTSTRING$(32,"*");
640 PRINT"***";TAB(13);"PUZZLE";
TAB(29);"***";
645 PRINT"***";TAB(5);"COLOR COM
PUTER VERSION";TAB(29);"***";
650 PRINTSTRING$(32,"*")
655 PRINT:PRINT
660 PRINTTAB(15);"BY:"
665 PRINTTAB(10);"DENNIS TEAGUE"
670 PRINT@482,"PRESS <ENTER> TO
CONTINUE  ";
675 LINE INPUT A$
680 CLS
685 PRINT"PUZZLE";TAB(26);"PAGE
2";
690 PRINT:PRINT:PRINT:PRINT
695 PRINT"THE OBJECT OF THIS PRO
GRAM      IS TO FIND ALL OF THE
WORDS     THAT YOU HAVE TOLD THE
 COMPUTER TO HIDE (UP TO 9).  TH
E WORDS   MAY BE 9 OR LESS LETTE
RS IN     LENGTH."
700 PRINT@482,"PRESS <ENTER> TO
CONTINUE  ";
705 LINE INPUT A$
710 CLS
715 PRINT"PUZZLE";TAB(26);"PAGE
3";
720 PRINT
725 PRINT"STEP #1          ENTER THE
 WORDS                     THAT YOU
WANT                       TO FIND."
730 PRINT"STEP #2          PRESS <EN
TER>                       WHEN DONE
 ENTERING                  WORDS (IF
 LESS                      THAN 9)."
735 PRINT"STEP #3          USING THE
 ARROW                     KEYS, MOV
E THE                      CURSOR TO
 THE                       FIRST LET
TER IN                     ONE OF TH
E WORDS."
740 PRINT@482,"PRESS <ENTER> TO
CONTINUE  ";
745 LINE INPUT A$
750 CLS
755 PRINT"PUZZLE";TAB(26);"PAGE
4";
760 PRINT:PRINT:PRINT
765 PRINT"STEP #4          WHEN ON T
HE FIRST                   LETTER, P
RESS THE                   <ENTER> K
EY."
770 PRINT"STEP #5          THEN PRES
S THE                      DIRECTION
 THE                       WORD IS G
OING                       WITH THE
UP-ARROW                   OR THE RI
GHT-ARROW.";
775 PRINT@482,"PRESS <ENTER> TO
CONTINUE  ";
780 LINE INPUT A$
785 CLS
790 PRINT"PUZZLE";TAB(26);"PAGE
5"
```

```
795 PRINT:PRINT
8ØØ PRINT"STEP #6              THEN PRES
S THE                          NUMBER OF
 LETTERS                       IN THE WO
RD (IF                         THE COMPU
TER FINDS                      A MATCH I
T SHOWES                       IT IN INV
ERSE                           VIDEO)."
8Ø5 PRINT"STEP #7              REPEAT ST
EPS #3-6                       FOR THE R
EST OF                         THE WORDS
."
81Ø PRINT@482,"PRESS <ENTER> TO
CONTINUE  ";
815 LINE INPUT A$
82Ø CLS
825 PRINT"PUZZLE";TAB(26);"PAGE
6";
83Ø PRINT:PRINT:PRINT
835 PRINT"FINAL NOTE:  WHEN IN D
OUBT          CHECK THE BOTTOM LINE
OF THE        SCREEN FOR INSTRUCTION
S."
84Ø PRINT@484,"PRESS <ENTER> TO
START   ";
845 LINE INPUT A$
85Ø CLS
855 RETURN
86Ø FOR A=1 TO 1ØØØ:NEXT A
865 CLS
87Ø PRINT"PUZZLE";TAB(26);"PAGE
8"
875 PRINT:PRINT
88Ø H=Ø:M=Ø
885 PRINT"PUZZLE FINISHED IN:"
89Ø IF S>36ØØ THEN H=H+1:S=S-36Ø
Ø:GOTO 89Ø
895 IF S>6Ø THEN M=M+1:S=S-6Ø:GO
TO895
9ØØ IF H=Ø THEN 915
9Ø5 PRINTH;"HOUR";
91Ø IF H>1 THEN PRINT"S" ELSE PR
INT
915 IF M=Ø THEN 93Ø
92Ø PRINTM;"MINUTE";
925 IF M>1 THEN PRINT"S" ELSE PR
INT
93Ø PRINTS;"SECOND";
935 IF S>1 THEN PRINT"S" ELSE PR
INT
94Ø PRINT:PRINT
945 PRINT"DO YOU WISH TO RUN THI
S PROGRAM AGAIN (Y/N)";
95Ø INPUT A$
955 IF LEFT$(A$,1)="Y" THEN CLS:
RUN 7Ø
96Ø PRINT:PRINT"THANKS FOR RUNNI
NG 'PUZZLE'"
965 END
```

GAME

4K

Alpine Slopes

By Mark Sabbatini

The 4K Color Computer is all but ignored today. There is almost no software written or available for it and magazines pay little, if any, attention to it. *Alpine Slopes* is an action game that fits into 4K of memory. There are many programming techniques used to fit the program into 4K that should be helpful to beginning programmers.

When you first run the program, a title screen appears showing the present highest score and the score of the last game played. Upon pressing a key, you are asked several questions on what type of course you want to ski. The difficulty level allows you to start at zero, 200 and 400 yards down the course, depending on the level set. The fast or slow option can only be used if your computer can handle the POKE 65495,0. Otherwise, leave it out of the program. The third question determines what type of race you are running and your score is kept accordingly. The fourth question determines how far down the slope you can see.

After the fourth prompt, an orange block appears at the top of the screen. Use the left- and right-arrow keys to move the skier back and forth. There's no need to press the keys repeatedly, just hold them down. As you move down the slope different objects appear. The most common of these are trees. They start out as bushes, but continue to get larger. Also, as you move down the course your distance is marked every 10 yards. You must avoid these yard markers.

Helpful objects include blue ski jumps, which move the skier ahead 50 yards. If you are racing slalom, gates will appear. Every gate you go through adds 100 points to your score. Once in a while the program will not give credit for hitting a slalom gate or a jump. Try to hit them head on to increase your chances of success. Once you run into something other than a ski jump or gate, the game is over.

Once the game is over, your yard total is added to the score one yard at a time. In a downhill race you will receive two points for every yard gained. If you want to skip the totaling process, press a key and the title screen appears. If your machine cannot handle the POKE here, leave it out. If you have the high score, it is indicated in reverse video. Press any key for another game.

Program Techniques

One of the most important requirements in an action game is speed. But in a game like this, there are often up to 15 or 20 objects on the screen at the same time. How does a simple program such as this manage to attain its speed? The answer is simple and often overlooked. The screen scrolls up every time it is filled. By printing 31 black spaces at the bottom of the screen, the program simulates movement.

One of the biggest aggravations I suffer in some programs is that after the game is over, I often accidentally press a key, thus causing the title screen or other to be missed. I avoided this in *Alpine Slopes* by putting in extra INKEY$ statements to help guard against accidental input.

The breakdown of the program lines is as follows:

130	—	Set high score to zero
140-300	—	Initialize game
340	—	Check for keyboard input
350	—	Position skier
360	—	Print skier
370	—	Check to see if skier hit jump or gate and give appropriate result
380	—	Check to see if skier hit tree or yard marker
390	—	Print tree
410	—	Random action to put gate or jump on screen
420	—	Scroll screen
430	—	Print yard marker
440	—	Add one to yard score
450	—	Determine tree size
480	—	Erase skier
490	—	Repeat loop
500-520	—	Game over and title screen
530-540	—	Set up new game

I think *Alpine Slopes* shows part of what can be done with 4K of memory. I hope this article inspires some 4K programmers. If you have any questions you may call or write me at 12139 Circle Drive, Conifer, CO 80433, (303) 838-7042. Please include an SASE when writing. □

The listing: ALPSLOPE

```
380 ......123
506 ......170
END ......36
```

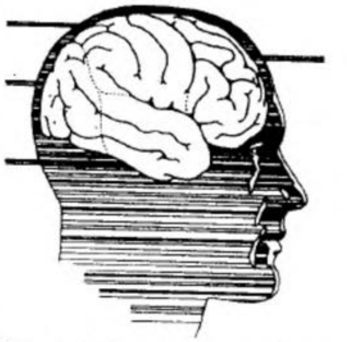
```
13Ø HI=Ø
131 GOTO51Ø
14Ø CLSØ:Q=28:H=1:FORA=1 TO31:Y$
=Y$+CHR$(128):NEXT:P$=CHR$(175):
FORA=1TO2:P$=P$+P$:NEXT:SL=1:S=Ø
:N=Ø
141 IFQQ=2THENN=2ØØELSEIFQQ=3THE
NN=4ØØ
24Ø L$=CHR$(8):R$=CHR$(9)
26Ø J$(1)=CHR$(215)+CHR$(219):J$
(2)=CHR$(215)+CHR$(223)+CHR$(219
):J$(3)=CHR$(215)+CHR$(223)+CHR$
(223)+CHR$(219):J$(4)=CHR$(215)+
CHR$(223)+CHR$(223)+CHR$(223)+CH
R$(219)
3ØØ A=15:B=3Ø:M=1:C$=CHR$(128)
34Ø IF(PEEK(343)AND8)=ØTHENX$=L$
ELSEIF(PEEK(344)AND8)=ØTHENX$=R$
ELSEX$=CHR$(133)
35Ø IFX$=L$ANDA>ØTHENZ$=CHR$(246
):A=A-1:B=B-2ELSEIFX$=R$ANDA<3ØT
HENZ$=CHR$(249):A=A+1:B=B+2 ELSE
Z$=CHR$(245)
36Ø PRINT@A,C$+Z$+C$;
37Ø IFPOINT(B+2,2)=3THENN=N+4Ø:E
=E+4Ø:FORL=24ØTO1STEP-2Ø:PRINT@4
8Ø,Y$:SOUNDL,1:NEXTL:GOTO39ØELSE
IFPOINT(B+2,2)=2THENS=S+1ØØ:SOUN
D15Ø,1:GOTO39Ø
38Ø IFPOINT(B+2,2)<>ØORPOINT(B+3
,2)<>ØTHEN5ØØ
39Ø PRINT@VI+RND(Q+1),J$(H);
41Ø P=RND(3Ø):IFP=1THENPRINT@VI-
1+RND(26),P$;ELSEIFP<7 ANDSR=1 T
HENPRINT@VI-1+RND(32),CHR$(159);
42Ø PRINT@48Ø,Y$
43Ø IFN/1Ø=INT(N/1Ø)THENPRINT@VI
+RND(Q),N;:FORT=1TO4Ø:NEXT:PRINT
@48Ø,Y$
44Ø N=N+1
45Ø IFN>199THENH=2ELSEIFN>399THE
NH=3ELSEIFN>699THENH=4
48Ø PRINT@A,C$+Z$+C$;
49Ø GOTO34Ø
5ØØ FORX=1TO15:SOUNDRND(1Ø)*15,1
:SET(A+A+RND(3),RND(3),RND(8)):N
EXT
5Ø1 N=N-13:CLS:PRINT@139,"YARDS"
N:PRINT@267,"SCORE";S:FORXX=1TO5
ØØ:NEXT:QQ$=INKEY$
5Ø2 POKE65495,Ø:FORZZ=1 TON:IFSR
=1 THENS=S+1:N=N-1ELSES=S+2:N=N-
1
5Ø3 PRINT@144,N"     ":PRINT@272,
S:SOUND5Ø,1:IFINKEY$<>""THEN5Ø5
ELSENEXTZZ
5Ø4 POKE65494,Ø:FORXX=1 TO2ØØ:NE
XTXX:GOTO51Ø
5Ø5 S=S+N
5Ø6 IFSR=Ø THENS=S+N
51Ø POKE65494,Ø:CLS:PRINT@73,"al
pine slopes":PRINT@171,"SCORE"S:
IFS>HS THENHS=S:IV=1
511 IFIV=1THENPRINT@264,"high sc
ore";HS ELSEPRINT@264,"HIGH SCOR
E"HS
515 PRINT@389,"PRESS ANY KEY TO
PLAY"
```

```
52Ø IV=Ø:Q$=INKEY$
521 Q$=INKEY$:IFQ$=""THEN521
523 CLS:PRINT@133,"DIFFICULTY LE
VEL (1-3)"
524 QQ$=INKEY$
525 QQ$=INKEY$:IFQQ$=""THEN525
526 QQ=VAL(QQ$):IFQQ<1ORQQ>3THEN
525
527 PRINT@2ØØ,"(F)AST OR (S)LOW"
:QQ$=INKEY$
528 QQ$=INKEY$:IFQQ$=""THEN528EL
SEIFQQ$="F"THENPOKE65495,ØELSEIF
QQ$<>"S"THEN528
53Ø PRINT@261,"(S)LOLOM OR (D)OW
NHILL"
531 QQ$=INKEY$
532 QQ$=INKEY$:IFQQ$=""THEN532EL
SEIFQQ$="D"THENSR=Ø ELSESR=1
533 PRINT@322,"VISIBLITY (1- GOO
D  4- BAD)"
534 QQ$=INKEY$
535 QQ$=INKEY$:IFQQ$=""THEN535
536 WW=VAL(QQ$):IFWW<1OR WW>4THE
N535
537 IFWW=1 THENVI=448ELSEIFWW=2
THENVI=354ELSEIFWW=3THEN VI=224E
LSEVI=128
54Ø Y$="":P$="":GOTO14Ø
```

GAME

32K ECB

Brain Games



By Kraig Brockschmidt

A collection of five challenging and exciting programs combined into one, *Brain Games* requires 32K ECB for the full program. It can easily be altered to fit into a 16K machine with the deletion of a graphics display in one of the games, which has no effect on the program. The five games are *Towers of Hanoi, Jump 15, Flip-Over, Short-Term Memory Test* and *The Human Cannonball.* The first three are challenges for your insight; the last two are a test for your memory and a fun practice in trial and error.

Type in the program or load it from cassette using CLOAD and ENTER. If you need to make the 16K alterations, follow the steps below:

1) In Line 7, change the 65 after the ON. . .GOTO statement to read 69.
2) Delete the X(900) after the DIM statement in Line 0 (make sure you also delete the comma preceding it).
3) Delete lines 65-68 and Line 73.
4) Remove the PUT statement in Line 71.

When you run the program, a title screen with the five individual program names appears. The line that reads TOWERS OF HANOI will be blinking alternately black and green. If you want to play that game, hit the ENTER key. If not, hit either the up- or down-arrow keys and the blinking will move accordingly. When you get to the title of the game you want, simply hit the ENTER key and it will play that game. If the title HUMAN CANNONBALL is blinking, hit the down-arrow key and the blinking will move up to TOWERS OF HANOI. Likewise, if you're on TOWERS OF HANOI, hit the up-arrow key and it will move to HUMAN CANNONBALL.

Each subprogram has completely different instructions and objectives. All five are outlined separately as follows.

Towers of Hanoi: The object is to transfer the stack of different-sized discs from the first peg to the third peg. As you begin, the computer asks you to select the number of discs you wish to play with, from two to six. Keep in mind that the more discs you choose, the longer and harder the game will be. You can place any disc on any peg, provided the disc underneath (if there is one) is not smaller than the one you are attempting to place on top.

To move the discs, the computer asks FROM? TO? and awaits your input. The first number should be the number of the peg from which you want to move the top disc, and the second number is where that disc is to be placed. When entering these numbers, separate them with a comma and check for typing errors to avoid a REDO? message, which will ruin the display. If you get all the discs from Peg #1 to Peg #3, you win. The computer tells how many moves you made and the minimum possible for that number of discs, given by the simple formula (2↑N)-1, where 'N' is the number of discs.

If you get stuck, type in two zeros for your move. The computer asks PLAY AGAIN (Y/N). If your answer is 'N', the program returns to the title page.

Jump 15: The object here is to eliminate all but one of the dots in the pyramid by jumping any other dot. The game starts with drawing the screen display and 14 dots, in the numbered spaces 2-15, and an open spot in Space 1. It then asks you FROM? TO?, to which you'll respond by typing in the number of the dot you wish to jump with, a comma and the number of the space you want it to jump into. Any jumping must be done in a straight line; you can't jump more than one dot or off the board. Once a dot is jumped over by another, it's taken off the board.

If you noticed that there aren't any

DATA statements containing all the possible jump combinations, it's because of a simple twist of this puzzle. If you take the number of the space you're jumping from and add the number of the space you're jumping to, divide by two and drop all fractions, that is the number of the only possible space you can jump over in that line. While there are several routes to take, the last dot will always be in Space 13.

Flip-Over: This is the hardest of the first three games. To win, you must move the white dots on the left side to occupy the four spaces on the right side, which are occupied with either blue or red dots. You can move any white dot one space to the right or jump *one* dot, but white *always* moves to the right — it cannot move backwards. Likewise, the blue or red dots can only move to the left in the same manner.

Operation is fairly simple. To "pick up" the dot you want to move, hit the right- or left-arrow key until the small line under the bottom of the rectangular border is directly under the dot you want to pick up. Press the ENTER key and it's been picked up. Next, move the small line under the space you want to move it to (this space must be black) and hit the ENTER key again. If it's a legal move, the dot will be moved, otherwise, you'll hear a short, low tone, and you must pick up a dot again and continue playing.

If you get stuck, which is not uncommon, press the 'Q' key and the familiar PLAY AGAIN appears.

Short-Term Memory Test: This is simply to test your memory, and you can't win or lose. On start-up, you'll be asked to remember two numbers. When you're ready to have them flashed before you, hit the ENTER key. Two numbers will be shown quickly and you'll be asked to type in those numbers in the order they appeared on the screen. Be sure you put a space between each number and no space at the end or the computer will say you're wrong. If correct, you'll be told, then you have to remember one more number. If you're wrong, you still have two more chances to remember the right sequence. This keeps running until you miss the sequence three times in a row. The average amount of numbers memorized by a person is seven.

Human Cannonball: The object is to land the stunt man safely into the net by choosing the correct angle of elevation of the cannon so he'll fly just as far as the net, but not over- or undershoot it. If you're running the 32K version, you'll have to wait while it draws a circus tent and the audience before playing. After you wait once, however, you don't have to wait again, unless you BREAK the program and rerun it. If you have 32K and wish to speed it up, you can make the 16K alterations.

Once the display is drawn, an entire half of the PMODE 4 screen is stored in Array X. Contrary to the ECB manual, you don't need to DIM X(256,96) as the manual says, but you only need one array variable ((DIM X(1) would be one array variable) for every 40 pixels being stored. Instead of "DIMming" X(256,96), which is 24576 bytes of memory, you need only DIM one-fortieth of that, or 615. In the program I use 900 to be on the safe side, so there's no chance of getting a ?BS Error caused by not having enough array variables.

Once you type in an angle between one and 89 (decimals are allowed), you'll see a graphics display of the results. If you hit the net, you'll hear some peppy musical notes and are asked to PLAY AGAIN? If you missed, you'll be told how much you missed by and asked for another angle. If you missed by a negative number, it means the net was undershot; a positive number means it was overshot. In either case, type in a smaller or larger angle and ENTER it. Muzzle velocity of the cannon is 110 meters per second.

Please realize that the first three games are rather challenging and it may take awhile to solve them. If you decide you can't figure them out, I'll be happy to send you the solutions for $1 and an SASE. I can also send you listings of each and any of the five individual games which are self-running, for 25 cents each and an SASE. Finally, if typing is not your game, send me $4 and I'll mail you a cassette with several copies of the program on it. My address is 14024 152nd Ave. S.E., Renton, WA 98056. □

9	47
18	47
28	57
37	5
51	241
62	78
69	137
END	172

The listing: BRAINGME

```
Ø CLS:CLEAR35Ø:DIMH(3),L(15),P(1
5),N$(15),X(9ØØ):K=Ø:R=RND(-TIME
R):'16K users! Make all required
 changes before running
1 RESTORE:T=1:' BRAIN GAMES
        (C) 1985 BY
      KRAIG BROCKSCHMIDT
      14Ø24 152ND AVE SE
      RENTON, WA   98Ø56
2 CLSRND(8):PRINT@1Ø,"BRAIN GAME
S";:PRINT@35,"<C>1985 KRAIG BROC
KSCHMIDT";
3 PRINT@135," TOWERS OF HANOI ";
:PRINT@2Ø3," JUMP 15 ";:PRINT@26
6," FLIP-OVER ";:PRINT@326," SHO
RT-TERM MEMORY ";:PRINT@39Ø," HU
MAN  CANNONBALL ";
4 FORG=1TO5:READC(G),J(G),J$(G):
NEXT:DATA1,136,"towers of hanoi"
,2,2Ø4,"jump 15",3,267,"flip-ove
r",4,327,"short-term memory",5,3
91,"human  cannonball"
5 RESTORE:PRINT@J(T),J$(T);:A$=I
NKEY$:IFA$="^"THENT=T-1:IFT<1 TH
ENT=5:GOTO8
6 IFA$=CHR$(1Ø)THENT=T+1:IFT>5 T
HENT=1:GOTO8
7 IFA$=CHR$(13)THEN ON C(T) GOTO
9,28,39,59,65
8 GOTO3
9 CLS:RESTORE:FORT=1TO15:READA$:
NEXT
1Ø U=Ø:INPUT"HOW MANY DISCS (2-6
)";B:IFB<2 OR B>6 THEN1ØELSEFORX
=1TO6:FORZ=2TO3:A(Z,X)=Ø:NEXT:NE
```

```
XT
11 NL$=STRING$(5,CHR$(128))+CHR$
(245)+STRING$(5,CHR$(128)):FORC=
1TO6:READT,X:D$(C)=STRING$(X,CHR
$(T)):NEXT:DATA239,1,223,3,2Ø7,5
,191,7,175,9,159,11
12 CLSØ:FORY=5TO18:SET(11,Y,8):S
ET(33,Y,8):SET(53,Y,8):NEXT:FOR
X=ØTO63:SET(X,18,8):NEXT:PRINT@3
26,"1";:PRINT@336,"2";:PRINT@347
,"3";
13 S=1:FORZ=1TO6:IF7-Z>B THENA(1
,Z)=ØELSE A(1,Z)=S:S=S+1
14 BB(Z)=A(1,Z):NEXT:H(1)=7-B:H(
2)=6:H(3)=6
15 FORZ=1TO3:FORX=1TO6:READ Q(Z,
X):NEXT:NEXT:DATA 96,128,16Ø,192
,224,256,1Ø7,139,171,2Ø3,235,267
,117,149,181,213,245,277
16 FORX=1TO6:FORZ=1TO3:PO=Q(Z,X)
+5-INT(LEN(D$(A(Z,X)))/2):IF A(Z
,X)>Ø THENPRINT@PO,D$(A(Z,X));EL
SEPRINT@PO-5,NL$;
17 NEXT:NEXT
18 PRINT@448,STRING$(32,CHR$(128
));:PRINT@416,"FROM ? TO ";:INPU
TQ,W:IF Q>3 OR W>3 OR Q=W OR Q<1
 OR W<1 THEN 23 ELSE IF A(Q,H(Q)
)>A(W,H(W)) AND A(W,H(W))>Ø THEN
 23 ELSE IFA(Q,H(Q))=Ø THEN 23
19 IFH(W)=6 AND A(W,H(W))=ØTHEN
A(W,6)=A(Q,H(Q)):A(Q,H(Q))=Ø:H(Q
)=H(Q)+1:IFH(Q)>6 THENH(Q)=6:GOT
O23ELSE23
2Ø A(W,H(W)-1)=A(Q.H(Q)):A(Q,H(Q
))=Ø:H(Q)=H(Q)+1:IFH(Q)>6THENH(Q
)=6
21 H(W)=H(W)-1:IFH(W)<1THENH(W)=
1
22 FORC=1TO6:IFA(3,C)<>BB(C)THEN
23ELSENEXT:U=U+1:GOTO25
23 IFQ<>Ø AND W<>Ø THENU=U+1ELSE
IFQ=Ø AND W=ØTHEN27
24 PRINT@428,"      ":GOTO16
25 CLS:PRINT"YOU WIN!!!!":PRINT"
IT TOOK YOU";U;"TURNS":PRINT"MIN
UMUM TURNS POSSIBLE:";(2^B)-1:PR
INT:PRINT
26 GOSUB84:GOTO9
27 PRINT@384,"YOU'RE STUCK - YOU
 LOSE!!!":GOTO26
28 XZ$=STRING$(32,CHR$(128)):FOR
 T=1TO45:READ A$:NEXT:Z$=CHR$(19
6)+CHR$(2ØØ):X$=CHR$(128)+CHR$(1
28):FOR T=1TO15:L(T)=1:READ P(T)
:IF LEN(STR$(T))=2 THEN N$(T)="Ø
"+RIGHT$(STR$(T),1) ELSE N$(T)=R
IGHT$(STR$(T),2)
29 NEXT:DATA79,141,145,2Ø3,2Ø7,2
11,265,269,273,277,327,331,335,3
39,343
3Ø Q=Ø:L(1)=Ø:CLSØ:FOR T=1TO15:P
RINT@P(T)+32,N$(T);:NEXT:FOR X=3
2TO55:Y=X-31:SET(X,Y,3):SET(63-X
,Y,3):NEXT:FOR X=9TO55:SET(X,24,
3):NEXT
31 FORT=1TO15:IFL(T)=1THENPRINT@
P(T),Z$;:NEXT ELSE PRINT@P(T),X$
;:NEXT
32 IFQ=1THENRETURNELSEPRINT@416,
"";:INPUT"FROM ? TO ";A,B:IF A>1
5 OR B>15 THENPRINT@429,"
";:PRINT@448,XZ$;:GOTO32
33 IFA=Ø AND B=ØTHEN38ELSEIFL(A)
=Ø OR L(B)=1 OR L(INT(A+B)/2)=ØT
HENPRINT@429,"          ";:PRINT@44
8,XZ$;:GOTO32
34 IFA=11 AND B=15THENPRINT@429,
"      ";:PRINT@448,XZ$;:GOTO32
35 L=Ø:L(A)=Ø:L(INT((A+B)/2))=Ø:
L(B)=1:FORT=1TO15:IFL(T)=1THENL=
L+1
36 NEXT:PRINT@429,"          ";:PRIN
T@448,XZ$;:IFL>1THEN31ELSEQ=1:GO
SUB31:PRINT@448,"YOU WIN!!!
    "
37 GOSUB84:FORT=2TO15:L(T)=1:NEX
T:GOTO3Ø
38 PRINT@448,"SO YOU'RE STUCK -
YOU LOSE!":GOTO37
39 PMODE4:PCLS:SCREEN1,1:PMODE3:
COLOR3,1:LINE(3Ø,76)-(226,116),P
SET,BF
4Ø FORX=47TO212STEP2Ø:CIRCLE(X,9
6),8,1,.8:PAINT(X,96),1,1:NEXT
41 FORX=47TO1Ø7STEP2Ø:PAINT(X,96
),Ø,3:NEXT:FOR X=147TO2Ø7STEP2Ø:
PAINT(X,96),2,3:NEXT:H=127
42 GOSUB54:GOSUB5Ø
43 X1=H
44 GOSUB5Ø
45 X2=H:GOTO51
46 Z=H:A$=INKEY$:IFA$=CHR$(8)THE
NH=H-2ØELSEIFA$=CHR$(9)THENH=H+2
ØELSEIFA$="Q"THEN57
47 IFH<47THENH=47ELSEIFH>2Ø7THEN
H=2Ø7
48 IFZ=H THENRETURN
49 LINE(Z-4,12Ø)-(Z+4,12Ø),PRESE
T:LINE(H-4,12Ø)-(H+4,12Ø),PSET:R
ETURN
5Ø GOSUB46:IFA$=CHR$(13)THENRETU
RNELSE5Ø
51 IFX1>X2 AND PPOINT(X1,96)=8TH
ENSOUND1,1:GOTO42ELSEIFX2>X1 AND
 PPOINT(X1,96)=6THENSOUND1,1:GOT
O42
52 IFABS(X1-X2)>4ØTHENSOUND1,1:G
OTO42
53 IF PPOINT(X2,96)<>5 THENSOUND
```

```
1,1:GOTO42ELSEC=PPOINT(X1,96):PA
INT(X2,96),C,3:PAINT(X1,96),1,3:
GOTO42
54 FORT=47TO1Ø7STEP2Ø:IFPPOINT(T
,96)=6THENNEXT:GOTO55ELSERETURN
55 FORT=147TO2Ø7STEP2Ø:IFPPOINT(
T,96)=8THENNEXT:GOTO56ELSERETURN
56 CLS:PRINT"YOU WIN!!!!!":GOTO5
8
57 CLS:PRINT"YOU'RE STUCK - YOU
LOSE!!!"
58 PRINT:PRINT:GOSUB84:GOTO39
59 FORT=1TO6Ø:READA$:NEXT:DATA "
R8ØD12ØL8ØU12ØR8ØM-8Ø,12Ø","BR4Ø
D12Ø","R8ØD6ØL8ØD6ØR8Ø","R8ØD6ØL
5ØR5ØD6ØL8Ø","D6ØR8ØL2ØU6ØD12Ø",
"R8ØL8ØD6ØR8ØD6ØL8Ø","D12ØR8ØU6Ø
L8Ø","R8ØM-8Ø,12Ø","R8ØD6ØL8ØU6Ø
D12ØR8ØU6Ø","R8ØD12ØU6ØL8ØU6Ø"
6Ø WW=Ø:FORA=ØTO9:READA$(A):NEXT
:PMODE2:PCLS:S=2
61 PCLS:CLS:PRINTS"NUMBERS":LINE
INPUT"HIT ENTER TO BEGIN ";D$:SC
REEN1,1:AN$="":FORT=1TOS:N=RND(1
Ø)-1:PCLS:DRAW"BM88,35;"+A$(N):A
N$=AN$+STR$(N):FORG=1TO25Ø:NEXTG
,T:INPUT"NUMBER SEQUENCE";SQ$:IF
SQ$=RIGHT$(AN$,S+(S-1))THEN64
62 IFWW=2THEN63ELSEPRINT"WRONG,
TRY AGAIN":WW=WW+1:FORT=1TO1ØØØ:
NEXT:GOTO61
63 PRINT"SORRY, GAME'S OVER.  HO
WEVER,   YOU DID REMEMBER";S-1;"
NUMBERS!":PRINT:GOSUB84:RESTORE:
GOTO59
64 FORT=1TO3:PRINT@236,"CORRECT"
:FORP=1TO1ØØ:NEXT:PRINT@236,"
     ":FORP=1TO7Ø:NEXTP,T:S=S+1:W
W=Ø:GOTO61
65 IFK=1THEN69ELSEPRINT@482,"4Ø
SECONDS TO SET THE SCREEN";
66 LINE-(Ø,3Ø),PRESET:PMODE4:PCL
S:PMODE3:COLOR3,Ø:FORX=ØTO255:Y=
3Ø*ABS(COS(X/27.1)):LINE-(X,Y),P
SET:IFX/2Ø=INT(X/2Ø)THENLINE-(X,
7Ø),PSET:LINE-(X,Y),PSET:NEXTELS
ENEXT
67 LINE(Ø,11Ø)-(255,11Ø),PSET:LI
NE(Ø,7Ø)-(255,7Ø),PSET:FORX=1ØTO
255STEP2Ø:T=T+1:IFT/2=INT(T/2)TH
ENC=2ELSEC=4
68 PAINT(X,5Ø),C,3:NEXT:PAINT(12
8,96),3,3:FOR X=ØTO255STEP1Ø:R=.
4:FORY=75TO1Ø5STEP1Ø:CIRCLE(X,Y)
,4,RND(4),R:R=R+.15:NEXTY,X:GET(
Ø,Ø)-(255,112),X,G:K=1
69 N$=";C3BR6U6D2L12U2D6E4F4E4D4
H4G4H4":XN=RND(2ØØ)+4Ø:CLS
7Ø PRINT"THE NET IS";XN;"METERS
AWAY.":INPUT"ANGLE OF CANNON(1-8
9)",A:IF A>89 OR A<1 THEN PRINT"
CAN'T DO THAT ANGLE!":GOTO7Ø
71 V=11Ø:V1=V:Y=18Ø:X=Ø:A1=A*.Ø1
754:PMODE4:PCLS:PMODE3:PUT(Ø,Ø)-
(255,112),X,PSET:DRAW"BM"+STR$(X
N)+",18Ø"+N$:LINE(Ø,191)-(Ø,191)
,PRESET:COLOR4,Ø:N=5:O=2:PLAY"L2
55"
72 COLOR2,Ø:LINE(Ø,18Ø)-(255,191
),PSET,BF:S=1:Z=114
73 LINE(Ø,Z)-(255,Z),PSET:Z=Z+S:
S=S*1.5:IF Z>18Ø THEN 74 ELSE 73
74 PMODE4:SCREEN1,1:PMODE3
75 COLOR4,Ø:LINE(Ø,174)-(6,18Ø),
PSET,BF:PLAY"L255O2CDO3CDO1CDO2C
D"
76 LINE-(X,Y),PSET:Y1=Y:X=X+(V1*
SIN(A1*2)/1Ø):Y=Y-(V*SIN(A1)/1Ø)
:V=V-1Ø:IF Y<Ø THEN Y=Ø ELSE IF
Y>18Ø THEN X=X-(V1*SIN(A1*2)/1Ø)
:GOTO82
77 IF Y<Y1 THEN N=N+1 ELSE N=N-1
78 IF N>12 THEN N=1:O=O+1:IF O>5
 THEN O=5
79 IF N<1 THEN N=12:O=O-1:IF O<1
 THEN O=1
8Ø PLAY"O"+STR$(O)+";"+STR$(N)
81 GOTO76
82 CLS:IF X<XN+6 AND X>XN-6 THEN
 PLAY"O5L1ØØCDEFGFEDCDEFGFEDCDEF
G":FOR T=1TO5ØØ:NEXT:PRINT"YOU G
OT IT!!!":GOTO 83 ELSE PLAY"L255
O1CDBBEBBFBBGBBCBB":FOR T=1TO5ØØ
:NEXT:PRINT"YOU MISSED BY ";-INT
(XN-X);"METERS":GOTO7Ø
83 GOSUB84:GOTO69
84 INPUT"PLAY AGAIN (Y/N)";A$:IF
 A$<>"N" THEN RETURN ELSE 1
```

One-Liner Contest Winner . . .

This one-liner is called *Pulsar*. It creates an appealing image for the eyes as circles appear to pulsate in and out of the screen. (On some computers you may have to RUN the program a second time because of the PCLEAR bug.

Mike Hall
Hartland, WI

The listing:

```
1Ø PCLEAR8:FORX=1TO8:PMODEØ,X:PC
LS:SCREEN1,1:FORY=X*2TO18ØSTEP16
:CIRCLE(128,96),Y:NEXTY,X:FORC=1
TO9999:D=(RND(7)-1)^2:FORX=1TO8:
PMODEØ,X:SCREEN1,1:FORZ=1TOD:NEX
TZ,X:FORX=8TO1STEP-1:PMODEØ,X:SC
REEN1,1:FORZ=1TOD:NEXTZ,X,C
```

SLEEP TIGHT

Your CoCo Is Awake Tonight

In this first of a two-part series, you'll learn how to use CoCo to keep an eye on the home front while you're away

By Dennis H. Weide

This month, I'll show you how to sleep tight when away from home, because your CoCo will be awake. This is a program and hardware project that allows you to scan your house or apartment from a remote location.

What are the advantages of this? The next time you go out of town, a housesitter can call your house at scheduled intervals to check security. If there's a terminal available, you can call and check yourself. The housesitter can leave messages to inform you of any major problems (i.e., the washing machine hose just broke and your house is flooded).

I always call my house and scan the alarms and messages at least once a day. It's a good feeling knowing the house is secure. Before you complain about long distance costs, they're really fairly inexpensive. Using an AT&T credit card and a Model 100, I can call from a motel room and not have to pay the motel charge for placing the call; an hour's worth of calls costs only $10 per month — a small price to pay for knowing your home is secure. I stay on the line less than 10 minutes per call.

There is one thing that must be understood before we go any further: The program will only scan alarms upon command. If someone tries to break in, the alarm will be tripped and stay tripped until reset, but the computer will not recognize the alarm until it has been commanded to scan the circuit.

A Look at the Software

First let's take a look at the program required to scan the alarms. Listing 1 is the alarm program. It's menu driven and requires a password to sign onto the system. You can change the password (up to 200 alphanumeric characters) by changing PASSWORD in Line 7900. If the correct password isn't given, the system won't allow access to any other program section.

Once signed on, the program asks for the date, time and terminal width. Date and time are for disk reference only. The system doesn't have a real-time clock. The width of the terminal you're calling from determines screen formatting. Default is 32 characters wide. The menu is displayed at the beginning of the program and anytime the ENTER key is pressed.

The terminal program used is a modified version of Dan Downard's terminal program from the November 1983 RAINBOW. The modifications include changing the memory location and disabling the BREAK key. This prevents anyone from having access to the disk by interrupting the program. It is hand assembled (without an editor/assembler), so there is no assembly language listing. To make it simple, it's in BASIC so it can be loaded and run from one program.

Reading and writing messages is easy.

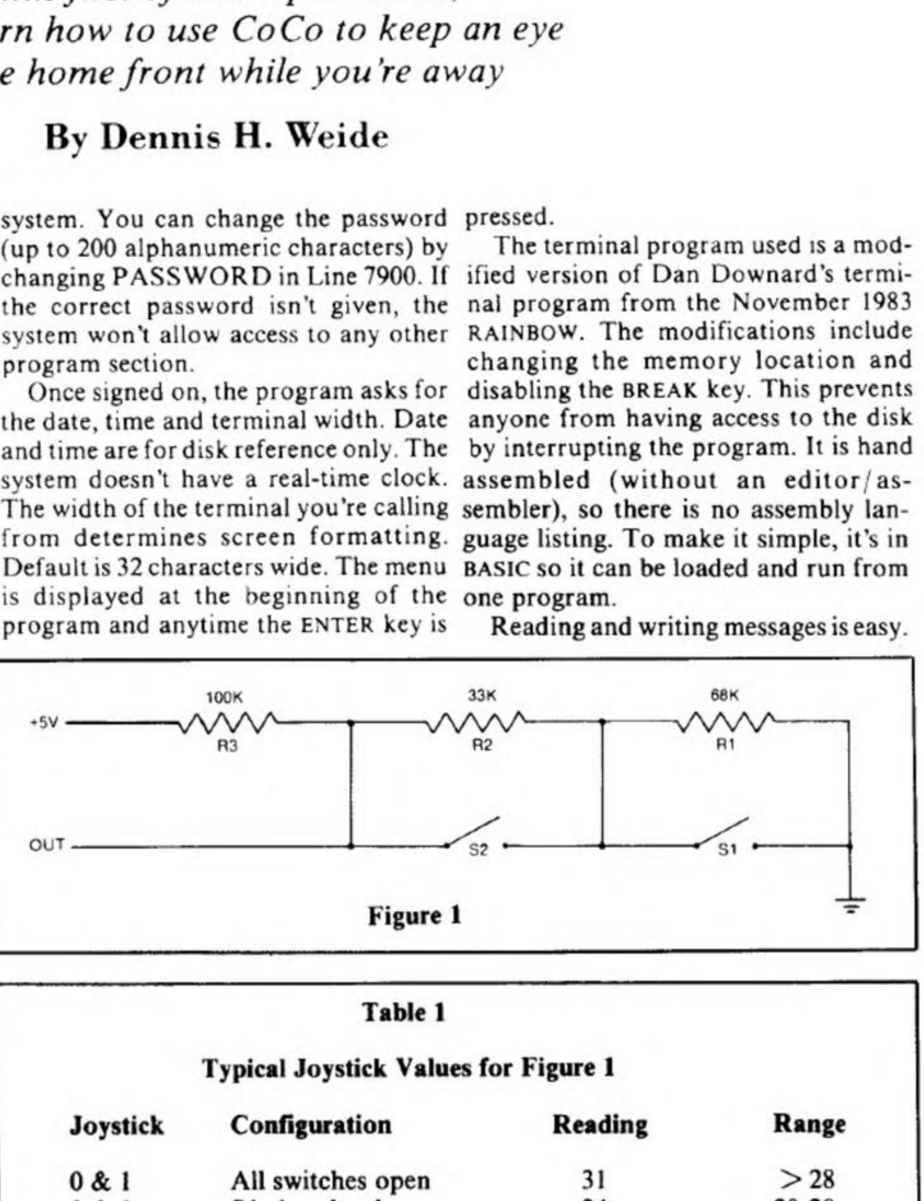


Figure 1

Table 1

Typical Joystick Values for Figure 1

Joystick	Configuration	Reading	Range
0 & 1	All switches open	31	>28
0 & 1	S1 closed only	24	20-28
0 & 1	S2 closed only	12	9-15
0 & 1	S1 & S2 closed	0	0
2 & 3	All switches open	30	>28
2 & 3	S1 closed only	23	20-28
2 & 3	S2 closed only	12	9-15
2 & 3	S1 & S2 closed	0	0

It uses DSKI$ to read the directory so wrong filenames won't stop program execution. Duplicate message names are not allowed since the second message would erase the first. To keep messages short, use abbreviations whenever possible. The maximum message length is 200 characters. The time you signed on is added to the message before being written to disk. When messages are read, the date and time are displayed at the beginning of each message. Erase messages via DOS using the KILL command.

Scanning Alarms

To activate the alarm system (turn it on), turn on the power supply switch and computer and load the program. When you run the program, sign on answering all prompts and request RESET ALARM to initialize the alarm circuits.

To scan the alarm circuit, request an ALARM STATUS from the main program menu. If the alarm system is turned off, an ALARM DEACTIVATED message appears and the program stops scanning. If the alarm system is on and no alarms exist, an ALARM ACTIVATED message is printed on the screen. If an alarm switch is open, scanning the alarm circuit will detect the alarm.

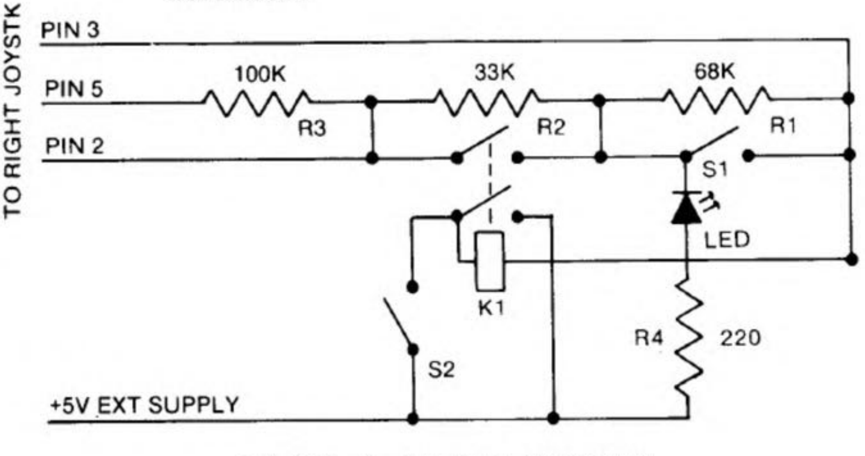
The circuit is divided into four sectors: north, south, east and west. A message is printed whenever an alarm condition exists showing which side of the house it's on. To check if the alarm is clear, use the RESET ALARM command. If clear, an ALARM CLEAR message is printed when scanned. If not, the alarm is printed again.

There is a tamper switch located in the electrical box with the on/off key switch. Any attempt to defeat the alarm system at this box closes the tamper switch. The next time the circuits are scanned, a TAMPER ALARM message is printed on the screen. The only way to reset this alarm is to turn off the external power supply.

All responses to commands are written to disk along with the time you signed on the system. The alarm log can be reviewed by entering the command ALARM LOG. This will list all system activity. If the log gets too full, erase it with the CLEAR LOG command. The next time you read the log, the only entry will be LOG CLEARED and the date and time you signed on.

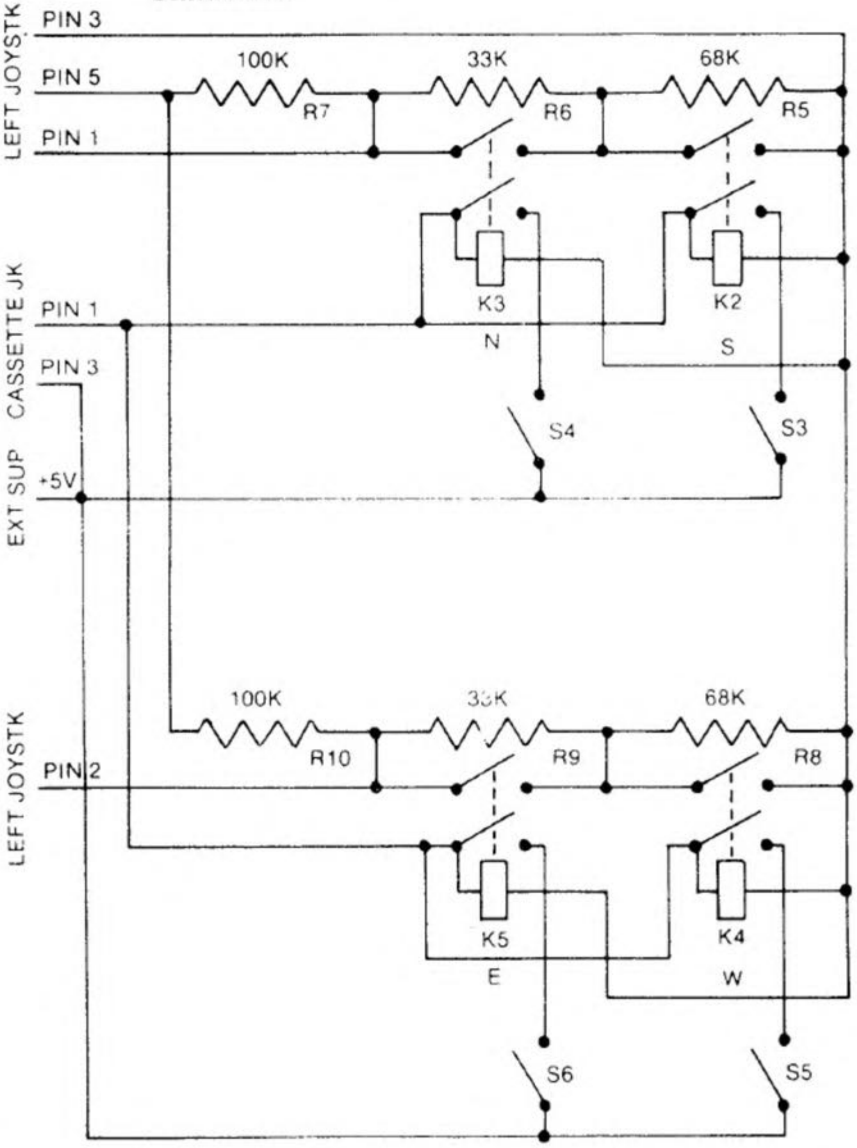
The last command is SIGN OFF. This, of course, signs you off the system and stops program execution until

Schematic 1



ON-OFF & TAMPER SWITCHES

Schematic 2



REMOTE ALARM LOOPS

ENTER is sent from the remote end or from the CoCo keyboard. The next time you sign on, press ENTER. The program jumps to the SIGN ON routine and you can begin again. Any command not acceptable will result in an INVALID COMMAND message.

A Look at the Hardware

The hardware for this program is easy to build. All input to the computer is via the joystick ports with the exception of the alarm reset circuit, which uses the cassette motor relay to reset the alarm circuit. Layout of the components is not critical and they can be soldered or wire wrapped on perf board.

A Simple Circuit

The basic circuit for the alarm system is a simple voltage divider acting as a variable resistor. Look at Figure 1 to see how it works. Three resistors are connected between +5 volts of the joystick port and ground. Closing either switch, S1 or S2, shorts the associated resistor and changes the voltage fed to the output. This output is fed to JOYSTK(0). Closing both switches puts the output at ground potential. Since the joystick port is the input to an analog-to-digital converter, voltage changes result in a corresponding change in the digital output of the converter. The digital output is then used to determine the alarm circuit status.

The entire alarm hardware works on this principle. Resistor R3 is used to ensure that the joystick port always sees at least 100K ohms of resistance. Anything less than that causes the other three joystick ports to vary erratically. Table 1 gives the values of the four joystick ports for all circuit configurations. Your readings will probably vary somewhat, but if they're within the range listed, you won't have to modify the program. Now let's look at the individual circuits to see what they do.

On/Off and Tamper Switches

Schematic 1 shows the on/off circuit and the tamper switch. You can see that the basic circuit of Figure 1 is used in this portion of the alarm circuit. Resistors R1 through R3, switch S1 and relay K1 make up the circuit. In addition, resistor R4 and a light emitting diode (LED) are used along with an external +5 volt supply to provide an alarm on/off indicator. All components are located on the alarm board except the on/off switch (S1), the tamper switch (S2) and the LED. These are located in the electrical box at the front door.

Table 2

Alarm Circuit Parts List

Item	Price	Radio Shack Part #
N/C window switches	$3.49 ea.	49-495
N/O tamper switch	1.39 ea.	49-528
Door lock switch	9.95 ea.	49-511
N/C 120 ft. foil	5.99 roll	49-502
N/C foil connectors	2.59 pkg/6	49-504
Joystick plugs (2 ea.)	1.19 ea.	274-020
Cassette plug (1 ea.)	1.49 ea.	274-003
5VDC DPDT mini-relay (5 ea.)	3.99 ea.	275-215
33K Ohm resistors (3 ea.)	.39 pkg/5	271-1341
68K Ohm resistors (3 ea.)	.39 pkg/5	271-1345
100K Ohm resistors (3 ea.)	.39 pkg/5	271-1347
220 Ohm resistor (1 ea.)	.39 pkg/5	271-1313
TLR-107 Hi-brite LED	.89 pkg/2	276-033
Hook-up and alarm wire	N/A	N/A

Schematic 3

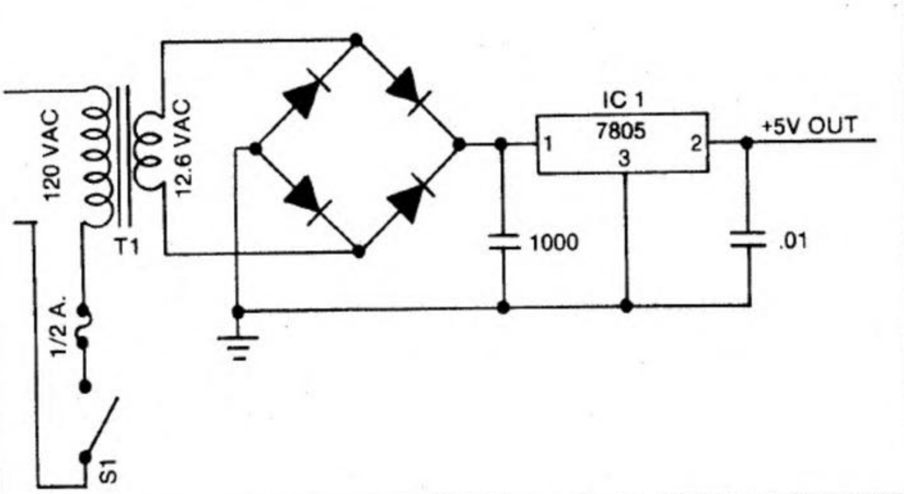


Table 3

Power Supply Parts List

Item	Price	Radio Shack Part #
12.6V Mini-transformer	$3.59 ea.	273-1365
1000 Mfd. elect. capacitor	1.59 ea.	272-1019
.01 Mfd. epoxy capacitor	.59 pkg/2	272-1065
Full wave bridge rectifier	.89 ea.	276-1161
7805 Fixed IC regulator	1.59 ea.	276-1770
Submini SPST toggle switch	1.59 ea.	275-612
Fuse holder	.99 ea.	270-367
Fuses	.69 pkg/3	270-1271
Project box for power Supply and alarm board	3.99 ea.	270-252

Closing the on/off switch (S1) turns on the alarm system by shorting R1. This lowers the voltage fed into the joystick port and the corresponding digital reading. This also places ground on the cathode of the LED, causing it to light. The LED is the on/off indicator at the front door.

The tamper switch (S2) is a push button switch that rests against the front plate of the electrical box at the front door. Any attempt to defeat the alarm system causes this switch to close momentarily operating relay K1. This relay shorts resistor R2, indicating to the computer that a tamper alarm exists. The relay has a self-holding path that keeps it operated until the power is removed. The tamper alarm can't be defeated from outside the house.

Remote Alarm Loops

The remote alarm loops are shown in Schematic 2. They are also similar to Figure 1 except they use two relays along with the switches. Again, these relays are used to hold the alarm activated until reset by program command. This prevents someone from opening a door or window and closing it before the alarm circuit can be scanned. Even though the door or window might be closed immediately, the relay stays released and the alarm condition remains until reset.

Switches S3 through S6 are the door and window switches located at each door and window. Each switch on the schematic consists of several switches in a series. The switch assignments are as follows:

S3 = All switches on south side of house in series
S4 = All switches on north side of house in series
S5 = All switches on west side of house in series
S6 = All switches on east side of house in series

The switches are closed when the door or window is closed. When the alarm circuit is turned on and a door or window is opened, that switch opens, releasing the associated relay and removing the short across its resistor. When scanned, the alarm circuit indicates an alarm in this sector. Once released, the relay can't be operated again until the RESET ALARM command is sent.

Notice the relays are all wired to Pin 1 of the cassette jack on the back of the computer. Pin 3 of the jack is wired to +5 volts. When the RESET command is given by the program, a relay inside the computer operates, shorting Pin 1 and Pin 3 of the cassette jack together, applying +5 volts to one side of each relay in Schematic 2. This momentarily operates the relays. They stay operated as long as the associated switches (S3-S5) are closed.

Notice the pin assignments on the schematic. Some wires connect to the left joystick, some to the right joystick, some to the cassette jack and some to an external +5 volt supply. This +5 volt supply is used so that no internal connections have to be made to the computer. Table 2 is a list of Radio Shack parts.

The External Power Supply

Schematic 3 shows the external power supply. It uses a 7805 regulator chip (IC 1) to provide a regulated +5VDC. Switch S1 is the on/off switch for the power supply and a ½ amp fuse is used in the transformer primarily for overload protection. The rectifier is an IC bridge.

Capacitor C1 is an electrolytic capacitor used to provide better regulation. Capacitor C2 is a bypass capacitor and is not absolutely required. Ground on the power supply must be connected to ground on either joystick port (Pin 3) to provide a common system ground. Table 3 gives a list of Radio Shack parts for the power supply.

Connecting to the Computer

Before connecting the alarm circuit to the computer, breadboard all the components and test your wiring, then verify that the voltages fed to the joystick port never drop below zero (negative) or rise above +4.8 volts for all possible circuit configurations. The analog-to-digital converter inside the computer can't handle voltages outside this range. Voltages outside this range indicate an error in the circuit wiring.

While the alarm circuit doesn't have an audible alarm, you can modify it to activate an alarm bell outside the house. The secret to a good alarm system is one nobody else can figure out. You can modify this circuit to provide all sorts of configurations and options.

For those who wish to save money, shop around for parts. You can save quite a bit if you mail-order parts or buy them from Radio Shack when they are on sale.

Next Month: Something for Those without Modems

Next month, I'll show you how to use this information to provide a local alarm system that uses a hardware clock and has continuous alarm scanning.

2900	27	13900	234
3800	14	16400	24
5900	139	18600	95
7900	79	21500	24
9800	251	END	88
12100	42		

The listing: RMTALARM

```
1ØØØ '      REMOTE BURGLAR ALARM
11ØØ '      BY DENNIS H. WEIDE
12ØØ '      142Ø1 MARQUETTE N.E.
13ØØ '      ALBUQUERQUE, NM  8712
3
14ØØ '      (C) 1985
15ØØ '
16ØØ '
17ØØ CLEAR2ØØØ,&H7ØØØ
18ØØ FOR X=&H7CC7 TO &H7D7F
19ØØ READ A:POKE X,A
2ØØØ NEXT X
21ØØ EXEC&H7CC7
22ØØ IF PEEK(&H7D8Ø)=134 AND PEE
K(&H7D81)=184 THEN PK=1
23ØØ FOR X=&H7D8Ø TO &H7EØ9
24ØØ READ A
25ØØ IF PK=Ø THEN POKE X,A
26ØØ NEXT X
27ØØ DATA 189,169,4Ø,142,4,32,16
,142,124,223,166,16Ø,167,128,16,
14Ø,125,127,35,246,57,125,128,Ø,
96,96,96,82,69,77,79,84,69,96,67
28ØØ DATA 79,67,79,96,66,85,82,7
1,76,65,82,96,65,76,65,82,77,96,
96,96,96,96,96,96,96,96,1Ø4,67,1
Ø5,96,113,121,12Ø,117,96,66,89,9
6
29ØØ DATA 68,11Ø,72,11Ø,87,69,73
,68,69,96,96,96,96,96,96,96,96,9
6,96,96,96,113,116,114,112,113,9
6,77,65,82,81,85,69,84,84,69,96
3ØØØ DATA 78,11Ø,69,11Ø,96,96,96
,96,96,96,96,96,96,96,96,96,96,9
6,65,76,66,85,81,85,69,82,81,85,
69,96,96,78,77,96,96,96,96,96,96
```

```
3100 DATA 96,96,96,96,96,96,96,9
6,96,96,96,96,96,96,96,96,120,11
9,113,114,115,96,96,96,96,96,96,
96,96,96,96,96,96,96,96,96
3200 DATA 134,184,151,230,190,1,
104,175,141,0,53,190,1,107,175,1
41,0,105,134,126,183,1,106,183,1
,103,48,141,0,11,191,1,104,48,14
1
3300 DATA 0,29,191,1,107,57,52,2
,13,111,38,14,129,13,38,7,134,10
,189,142,12,134,13,189,142,12,53
,2,126,125,168,15,112,13,111,38
3400 DATA 50,50,98,52,21,173,159
,160,0,39,2,32,36,246,255,34,86,
37,242,141,35,198,1,52,4,79,141,
26,246,255,34,86,36,2,170,96,104
3500 DATA 96,36,242,50,97,132,12
7,129,3,38,1,79,53,149,126,125,1
68,141,0,52,2,150,230,33,254,74,
38,251,53,130
3600 IF PK=0 THEN EXEC &H7D80
3700 FOR X=4 TO 9
3800 READ AM$(X)
3900 NEXT X
4000 DATA NORTH SIDE ALARM,SOUTH
 SIDE ALARM,NORTH/SOUTH ALARM
4100 DATA EAST SIDE ALARM,WEST S
IDE ALARM,EAST/WEST ALARM
4200 DIM DR$(68)
4300 GOTO 7100
4400 '
4500 '      START OF ALARM SCAN
4600 '
4700 FOR X=0 TO 3
4800 A(X)=JOYSTK(X)
4900 NEXT X:Y=4
5000 '
5100 '      COMPUTE ALARM VALUE
5200 '
5300 PRINT STRING$(TW,"*");
5400 IF A(1)>39 THEN AR$="ALARM
DEACTIVATED":GOSUB 20200:GOTO 65
00
5500 IF A(1)>9 AND A(1)<15 THEN
AR$="ALARM ACTIVATED":GOSUB 2020
0
5600 IF A(1)=0 THEN AR$="TAMPER
ALARM":GOSUB 20200
5700 IF A(1)>9 AND A(1)<15 THEN
AR$="TAMPER CLEAR":GOSUB 20200
5800 FOR X=2 TO 3
5900 IF A(X)=0 THEN AF(Y)=0:AF(Y
+1)=0:AF(Y+2)=0:FF=FF+1
6000 IF A(X)>9 AND A(X)<15 THEN
AR$=AM$(Y):GOSUB 20200:AF(Y)=1:F
1=1
6100 IF A(X)>20 AND A(X)<28 THEN
 AR$=AM$(Y+1):GOSUB 20200:AF(Y+1
)=1:F1=1
```

```
6200 IF A(X)=>28 THEN AR$=AM$(Y+
2):GOSUB 20200:AF(Y+2)=1:F1=1
6300 Y=Y+3:NEXT X
6400 IF F1=1 AND FF=2 THEN AR$="
ALARM CLEARED":F2=0::GOSUB 20200
:F1=0
6500 FF=0
6600 PRINT STRING$(TW,"*")
6700 RETURN
6800 '
6900 '      PASSWORD AND COMMANDS
7000 '
7100 CLS
7200 INPUT"ENTER PASSWORD";PW$
7300 TW=32
7400 PRINT:INPUT"ENTER DATE (MM/
DD/YY)";DT$
7500 PRINT:INPUT"ENTER TIME (HH/
MM)";CL$
7600 PRINT
7700 CL$=LEFT$(CL$,2)+":"+RIGHT$
(CL$,2)
7800 CL$=DT$+"  "+CL$
7900 IF PW$="PASSWORD" THEN AR$=
"SIGN ON":PRINT STRING$(TW,"#");
:GOSUB 20200 ELSE PRINT:GOTO 720
0
8000 PRINT STRING$(TW,"#")
8100 PRINT:INPUT"ENTER TERMINAL
WIDTH";TW:IF TW<20 THEN TW=32
8200 '
8300 '      COMMAND MODULE
8400 '
8500 CLS:PRINT:PRINT
8600 TW$="PROGRAM MENU"
8700 PRINTTAB((TW-LEN(TW$))/2) T
W$
8800 TW$="-----------
8900 PRINTTAB((TW-LEN(TW$))/2) T
W$
9000 PRINT
9100 TW$="1. WRITE MESSAGE"
9200 PRINTTAB((TW-LEN(TW$))/2) T
W$
9300 TW$="2. READ MESSAGE "
9400 PRINTTAB((TW-LEN(TW$))/2) T
W$
9500 TW$="3. ALARM STATUS"
9600 PRINTTAB((TW-LEN(TW$))/2) T
W$
9700 TW$="4. ALARM LOG    "
9800 PRINTTAB((TW-LEN(TW$))/2) T
W$
9900 TW$="5. RESET ALARM "
10000 PRINTTAB((TW-LEN(TW$))/2)
TW$
10100 TW$="6. CLEAR LOG     "
10200 PRINTTAB((TW-LEN(TW$))/2)
TW$
10300 TW$="7. SIGN OFF      "
```

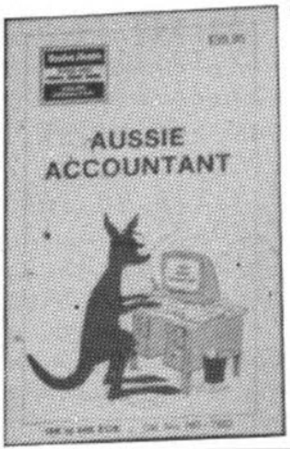
Tandy ELECTRONICS

Exciting Software for Your Color Computer

"Aust Quiz"

9^{95}

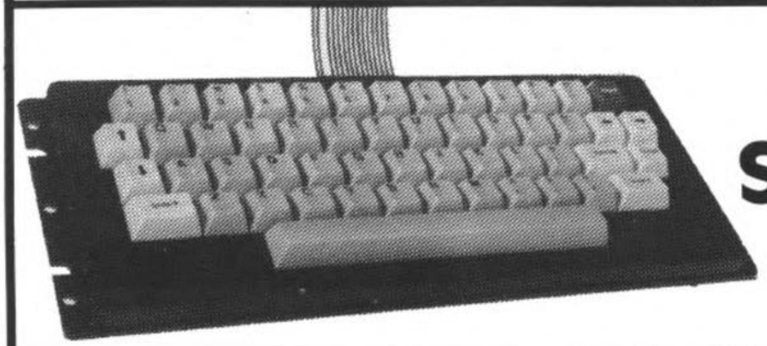
Test your local knowledge with this "brain teaser"! How much do you really know about Australian geography? A fun way to learn for people of all ages. Scoreboard shows progress, has 8 different programs. 26-7960



"Aussie Accountant"

39^{95}

The easy way to manage your finances and keep track of all accounts! Provides you with a double-entry record of income and expenditure plus monthly/ annual totals. Automatically balances forward to next fiscal year. 26-7962



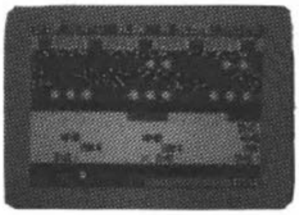
Low Profile Keyboard

Reg **59.95**

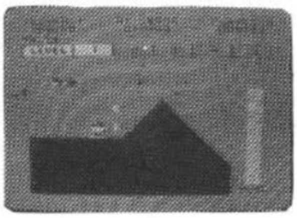
Save $^{\$}$20 39^{95}

Is your keyboard getting old? Upgrade with this current low profile design and SAVE $20! 26-3016

Tandy's Arcade-Style Fast-Action Games



Frogger. Guide frog across a busy highway and dangerous river to his home. 26-7304



Scramble. Your mission: destroy the enemy base! Beware of UFO's, missiles, maze. 26-7306

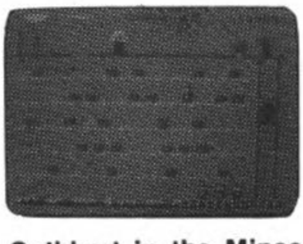
Eight Ball. Armchair snooker! Line up the cue, aim then sink the balls. 26-7311

Hours of Enjoyment

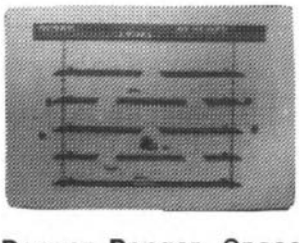
29^{95} EACH

Test your reflexes against the computer. May require joystick or cassette recorder.

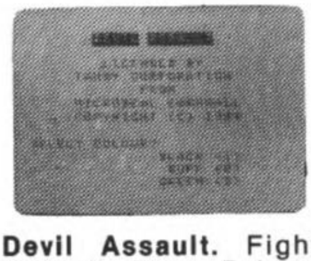
King Tut. Search for lost treasure in Tut's tomb! Look out for his ghost and snakes. 26-7313



Cuthbert in the Mines. Dodge rolling trucks and fireballs to escape the mines! 26-7314



Danger Ranger. Space adventure! Brave the Acid Chamber, demons, radioactive bats. 26-7315



Devil Assault. Fight Vampire Bats, Robots, deadly sproings, then the Devil! 26-7316

Easily Affordable Dot-Matrix Triple-Mode "Personal Printer"

Word Processing, Data Processing and Dot-Addressable Graphics

- ***Includes New Cursive and Microfont Styles***

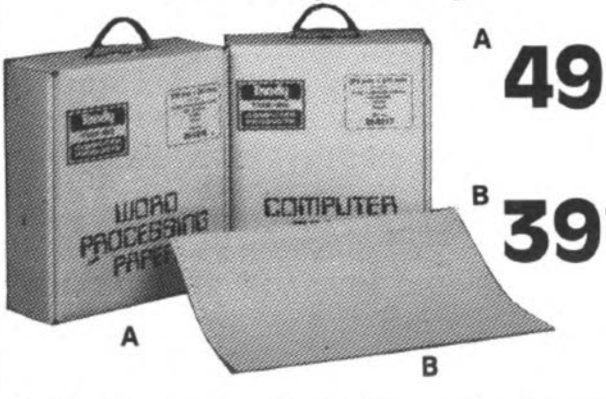
Save 30%

Reg **499.95**

349.95

DMP-110. Now you can afford high-performance printing for your system! WP mode has proportionally spaced or correspondence-quality characters for letters or reports. DP mode prints monospaced characters in a choice of 3 styles. Has underline, italic super, subscripts and microfonts. 26-1271

Stock Up on Paper

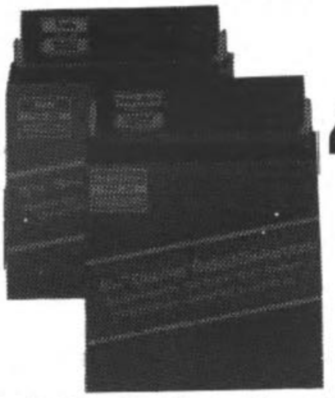


A **49.95**

B **39.95**

A. Wordprocessing. Blank white design. Size: 241 x 279mm. Tote of 900 sheets. 26-9316
B. Fanfold. Blue H/S. 241 x 279. 1000 sheets. 26-9317

13.3cm Diskettes



A **39.95**

B **49.95**

Unformatted, double-density, soft-sectored disks.
A. Single-Sided. 40 track. Box of 10. 26-406
B. Double-Sided. 40 track. Box of 10. 26-412

WE SERVICE WHAT WE SELL!

Tandy ELECTRONICS

A DIVISION OF TANDY
AUSTRALIA LIMITED
INC IN N S W

Nearly 350 Stores Australia-Wide

Available from
350 Stores Australiawide
including Tandy Computer Centres

**Independent Tandy Dealers may not be participating in this ad or have every item advertised. Prices may also vary at Individual Dealer Stores*

```
 1Ø4ØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
 1Ø5ØØ PRINT
 1Ø6ØØ LINEINPUT"GO >";CM$
 1Ø7ØØ PRINT
 1Ø8ØØ CM=VAL(CM$)
 1Ø9ØØ IF CM>Ø AND CM<9 THEN 118Ø
Ø
 11ØØØ IF CM$="" THEN 86ØØ
 111ØØ TW$="INVALID COMMAND"
 112ØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
 113ØØ GOTO 1Ø5ØØ
 114ØØ '
 115ØØ '      START PROGRAM HERE
 116ØØ '      USE "ON CM GOSUB"
 117ØØ '
 118ØØ ON CM GOSUB 146ØØ,173ØØ,47
ØØ,212ØØ,238ØØ,248ØØ,227ØØ
 119ØØ IF F5=1 THEN F5=Ø:GOTO 72Ø
Ø
 12ØØØ IF AR$="SIGN OFF" THEN PRI
NT:GOTO 72ØØ
 121ØØ GOTO 1Ø6ØØ
 122ØØ '
 123ØØ '      READ DIRECTORY
 124ØØ '
 125ØØ '
 126ØØ Z=1
 127ØØ TW$="CURRENT MESSAGES"
 128ØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
 129ØØ TW$="----------------
 13ØØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
 131ØØ FOR Y=3 TO 11
 132ØØ DSKI$ Ø,17,Y,A$,B$
 133ØØ FOR X=1 TO 128 STEP 32
 134ØØ IF MID$(A$,X,1)=>"A" AND M
ID$(A$,X,1)=<"Z" AND MID$(A$,X+8
,3)="MSG" THEN DR$(Z)=MID$(A$,X,
8):PRINTTAB((TW-LEN(DR$(Z)))/2)
 DR$(Z):Z=Z+1
 135ØØ IF MID$(B$,X,1)=>"A" AND M
ID$(B$,X,1)=<"Z" AND MID$(B$,X+8
,3)="MSG" THEN DR$(Z)=MID$(B$,X,
8):PRINTTAB((TW-LEN(DR$(Z)))/2)
 DR$(Z):Z=Z+1
 136ØØ NEXT X
 137ØØ NEXT Y
 138ØØ RETURN
 139ØØ '
 14ØØØ '      READ AND WRITE MSGS
 141ØØ '
 142ØØ '
 143ØØ '
 144ØØ '      WRITE MESSAGE
 145ØØ '
 146ØØ NA$="":A$="":PRINT:PRINT
 147ØØ PRINT"SPLIT MESSAGE PROPER
LY ON SCREENUSE NO MORE THAN 2ØØ
 CHARACTERS"
 148ØØ PRINT:PRINT
 149ØØ INPUT"ENTER MESSAGE NAME";
NA$:PRINT:IF NA$="" THEN RETURN
 15ØØØ NA$=NA$+"          ":NA$=LEF
T$(NA$,8)
 151ØØ GOSUB 126ØØ
 152ØØ FOR X=1 TO 68
 153ØØ IF NA$=DR$(X) THEN 156ØØ
 154ØØ NEXT X
 155ØØ GOTO 16ØØØ
 156ØØ PRINT:PRINT
 157ØØ TW$="MESSAGE NAME ALREADY
IN USE"
 158ØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
 159ØØ GOTO 149ØØ
 16ØØØ NA$=NA$+".MSG"
 161ØØ PRINT:PRINT"ENTER MESSAGE"
:PRINT
 162ØØ INPUT A$
 163ØØ IF LEN(A$)>19Ø THEN PRINT:
PRINT"warning 1Ø CHARACTERS REMA
INING"
 164ØØ A$=CL$+"  "+A$
 165ØØ OPEN"O",#1,NA$
 166ØØ WRITE#1,A$
 167ØØ CLOSE#1
 168ØØ PRINT
 169ØØ RETURN
 17ØØØ '
 171ØØ '      READ MESSAGE
 172ØØ '
 173ØØ GOSUB 126ØØ:PRINT
 174ØØ INPUT"ENTER FILE TO READ "
;NA$
 175ØØ NA$=NA$+"        "
 176ØØ NA$=LEFT$(NA$,8)
 177ØØ FOR X=1 TO 68
 178ØØ IF NA$=DR$(X) THEN 183ØØ
 179ØØ NEXT X
 18ØØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
 181ØØ TW$="NO MESSAGE BY THAT NA
ME"
 182ØØ RETURN
 183ØØ PRINT:PRINT:NA$=NA$+".MSG"
 184ØØ OPEN"I",#1,NA$
 185ØØ INPUT#1,A$
 186ØØ PRINT A$:PRINT
 187ØØ CLOSE#1
 188ØØ RETURN
 189ØØ '
 19ØØØ '      READ AND WRITE ALARM
S TO DISK
 191ØØ '
 192ØØ '
 193ØØ '
 194ØØ '      OPEN BUF #1
```

```
195ØØ '
196ØØ OPEN"D",#1,"ALARM.DAT",33
197ØØ FIELD#1,18 AS AL$,15 AS TI
$
198ØØ RETURN
199ØØ '
2ØØØØ '      WRITE ALARMS TO DISK
2Ø1ØØ '
2Ø2ØØ PRINTTAB((TW-LEN(AR$))/2)
AR$
2Ø3ØØ GOSUB 196ØØ
2Ø4ØØ LSET AL$=AR$
2Ø5ØØ LSET TI$=CL$
2Ø6ØØ PUT#1,LOF(1)+1
2Ø7ØØ CLOSE#1
2Ø8ØØ RETURN
2Ø9ØØ '
21ØØØ '      READ ALARMS FROM DIS
K
211ØØ '
212ØØ GOSUB 196ØØ
213ØØ PRINT
214ØØ FOR X=1 TO LOF(1)
215ØØ GET#1,X
216ØØ TW$="*****LOG ENTRY*****"
217ØØ PRINTTAB((TW-LEN(TW$))/2)
TW$
218ØØ PRINTTAB((TW-LEN(TI$))/2)
TI$
219ØØ PRINTTAB((TW-LEN(AL$))/2)
AL$:PRINT
22ØØØ NEXT X
221ØØ CLOSE#1
222ØØ RETURN
223ØØ '
224ØØ '      SIGN OFF
225ØØ '
226ØØ '
227ØØ AR$="SIGN OFF"
228ØØ PRINT STRING$(TW,"#");
229ØØ GOSUB 2Ø2ØØ
23ØØØ PRINT STRING$(TW,"#")
231ØØ FOR X=1 TO 3Ø
232ØØ PRINT STRING$(TW,"*");
233ØØ NEXT X
234ØØ INPUT X$:RETURN
235ØØ '
236ØØ '      RESET ALARM
237ØØ '
238ØØ MOTOR ON
239ØØ AR$="RESET ALARM"
24ØØØ PRINT STRING$(TW,"#");
241ØØ GOSUB 2Ø2ØØ
242ØØ PRINT STRING$(TW,"#")
243ØØ MOTOR OFF
244ØØ RETURN
245ØØ '
246ØØ !      ERASE ALARM LOG
247ØØ '
248ØØ KILL"ALARM.DAT"
249ØØ PRINT STRING$(TW,"#");
25ØØØ AR$="LOG CLEARED"
251ØØ GOSUB 2Ø2ØØ
252ØØ PRINT STRING$(TW,"#")
253ØØ RETURN
```

HAM UTILITY 16K ECB

COAX

by Irvin McNally

This program provides for the determination of the various factors related to a transmission line. The inputs are the input power, the length of the transmission line, characteristic impedance of the line, line attenuation at the desired frequency and the measured SWR at the input to the line. Using 100 as the input power (PI), the answers will be as percentages of the input power.

Coax computes and prints the output SWR (SO), the load resistance (ZR), the forward power (FP), the lòad power (PL), the reflected power (PR), the power to a matched load (PM) and the losses in Db due to the presence of standing waves. The program is completely self-prompting.

65 237
180 86
END 103

The listing: COAX

```
5 PRINT#-2, "DETERMINATION OF FA
CTORS FOR A GIVEN LENGTH OF TRAN
SMISSION LINE, CHARACTERISTIC IM
PEDANCE, LINE ATTENUATION AT THE
 DESIRED FREQUENCY AND THE MEASU
RED SWR AT THE INPUT TO THE TRAN
SMISSION LINE."
1Ø PRINT "WHAT IS THE INPUT POWE
R? EXPRESS AS 1ØØ AND THE ANSWER
S WILL BE AS PERCENTAGES OF THE
INPUT POWER."
15 INPUT "ENTER"; PI
2Ø PRINT#-2, PI "INPUT POWER (PI
)."
25 PRINT "WHAT IS THE IMPEDANCE
OF THE COAX OR TRANSMISSION LINE
?"
3Ø INPUT "ENTER"; ZO
35 PRINT#-2, ZO "IMPEDANCE (ZO)
OF COAX OR TRANSMISSION LINE."
4Ø PRINT "WHAT IS THE LENGTH OF
THE LINE IN FEET (L)?"
45 INPUT "ENTER"; L
5Ø PRINT#-2, L "LENGTH (L) OF TH
E LINE IN FEET."
55 PRINT "WHAT IS THE ATTENUATIO
N IN DB PER 1ØØ FEET AT THE DESI
RED FREQUENCY?"
6Ø INPUT "ENTER"; X
65 PRINT#-2, X "ATTENUATION (X)
IN DB PER 1ØØ FEET AT THE DESIRE
D FREQUENCY."
7Ø A=L*X/1ØØ
75 PRINT#-2, A "MATCHED LOAD ATT
ENUATION (A) OF LINE IN DB."
8Ø PRINT "WHAT IS THE SWR AT THE
 INPUT TO THE LINE (SI)?"
85 INPUT "ENTER"; SI
9Ø PRINT#-2, SI "INPUT SWR (SI).
"
95 M=SI-1
1ØØ N=SI+1
1Ø5 Q=N/M
11Ø R=A/1Ø
115 T=1Ø^R
12Ø U=T+Q
125 SO=U*(1/(Q-T))
13Ø PRINT#-2, SO "SWR AT OUTPUT
(SO)."
135 ZR=ZO*SO
14Ø PRINT#-2, ZR "LOAD RESISTANC
E (ZR)."
145 S=ZR/ZO
15Ø C=(S-1)/(S+1)
155 G=EXP(-.46*A)
16Ø H=EXP(-.23*A)
165 J=C^2
17Ø PF=PI/(1-(J*G))
175 PRINT#-2, PF "FORWARD POWER
(PF)."
18Ø PL=(1-J)*H*PI/(1-(J*G))
185 PRINT#-2, PL "LOAD POWER (PL
)."
19Ø PR=PF-PI
195 PRINT#-2, PR "REFLECTED POWE
R (PR)."
2ØØ PM=PI/1Ø^(A/1Ø)
2Ø5 PRINT#-2, PM "POWER TO A MAT
CHED LOAD (PM)."
21Ø LA=4.343*LOG(PM/PL)
215 PRINT#-2, LA "ADDITIONAL LOS
S IN DB DUE TO PRESENCE OF STAND
ING WAVES."
22Ø LT=A+LA
225 PRINT#-2, LT "TOTAL LOSS IN
DB IN THE PRESENCE OF STANDING W
AVES."
23Ø PRINT#-2, "PROGRAM PREPARED
FOR TRS-8ØC BY K6WX."
```

ASSEMBLY FILE

by Kevin

The New year is always presumed to be a time of taking stock and assesing your previous year so why not be conventional and decide what it is important to learn in your assembly programming.

But first let me tell you a little story. I served an apprenticship as an aircraft engineer maintaining DC8 and DC10 airliners. On the DC8 (1950's technology) the emphasis in our work and training was very much based around our developing a thorough knowledge of the aircraft's systems and then when a defect occured the LAME (Licensed Engineer) would use his knowledge to determine the faulty component and we would then proceed to replace that component. Naturally somtimes his "guess" would be wrong ,and we would go on replacing components until the system tested OK again and the plane certified airworthy. This system of maintenance is rather rudely called shotgun maintenance. On the DC10's (1970's technology) the training continued to emphasise a thorough understanding of the aircraft's systems but when a defect developed the LAME would go to a manual open the system page and follow a flowchart that told him tests to make using built in test equipment on the aircraft and the chart would thus tell him which components to replace. Of course this technology was far quicker and more reliable than the old shotgun system but just about anyone can read the manual and do the work so the LAME ends up simply filling out and signing forms. Ain't technology grand.

Now back to the subject. Assembly language programming can be a highly developed art or it can be little more than a learned proceedure. You have probably already gathered that my programming tends to be rather haphazzard with very little sign of refined proceedure. I argue that any professional programmer can be taught to program acceptably or superbly simply by following simple rules and proceedures. But I argue even further that true programming ability comes from developing a total understanding and familiarity with the system you are working.

So that leads to the question what is it important for you to learn to begin to develop the art of programming the CoCo. Well first of all it is not necessary to completely understand how to quickly convert from Binary to Decimal to Hex but is is necessary to know the relationship between these three froms of expressing the same value. You can always use tables to carry out the conversions for you and in fact the more you program and use assembly the more you become familiar with the relationship between and where these differnt forms have a use.

Likewise with the instruction set. You must have at hand a reference giving all the different forms of the instruction set for the chip you are programming, preferably including the Machine Language value for that instruction.

Lastly a good working knowledge of the architcture of the components and the relationship between the different components making up the hardware of your computer has enormous value in assembly programming. This includes some form of detailed memory map of the primary operating system you plan to work under (in our case we are only dealing with TANDY BASIC).

With all of this you are well prepared to begin to try and understand the actual programing. So how do we go about that?

I've said it before and I'll say it again. There is no better value to be had than working through Willian Barden Jnr's boook TRS-80 COLOR COMPUTER ASSEMBLY LANGUAGE PROGRAMMING (Tandy Cat-No. 62-20779. It is remarkably good value at $9.95. This together with the study of other peoples listings at every oportunity, trying to understand why they use the techniques they do. But don't get hung up on everyone elses work. I have been known to use and have discovered others using some pretty shoddy techniques based on the simple argument that it works!! If you don't think someone elses routine is up to scratch then don't hesitate. Fix it to your liking. You may even get lucky and discover that the original author did have an obscure reason for his technique, but usually there's no special reason.

As to your actual programming. It is incredibly difficult to learn anything if you don't have an adequate Assembly Program. And in my opinion there is very little that can be said against Tandy's EDTASM+ ROMPAC. By the time you are ready to be overly critical of any faults you may percieve in EDTASM then you will be certain to know exactly what you want out of an Assembler. Naturally enough those who run a disk system would be better off with DISK EDTASM.

Please learn to make use of the 6809's ability to produce very good Position Independant Code very simply. Ultimately it makes your work so much easier. And finally. With the large variety in versions of ROM that TANDY have produced it is doubly important that you get into the habit of using standard points of entry before making use of the routines within the BASIC ROM's. I have an absolutly marvelous program which gives my system 40K of user RAM but it is of use to only those limited numbers of people with the oldest version of CoCo. In the newer machines my program is useless simply because it breaks this rule.

Above all else programming should be enjoyable and if your results give you the satisfaction you need even though you may be using SHOTGUN methods then who wants to program by the book.

This simple project allows you to have a computer rap session

CoCo Conversation

By Larry Landwehr

Just about everyone knows computers can communicate with each other over telephone lines — all it takes is a modem and some software commonly known as a terminal package. Did you realize that computers can also talk directly with each other over a simple wire? By modifying the cable that connects the CoCo with a printer, two CoCos can talk together.

First, you need two computers (some people have an extra one lying around because of upgrades). If you don't maybe you can do this project with a friend.

Second, you need a terminal package. I used the one written by Richard Campbell from the March 1984 issue of *The Color Computer Magazine*. It has the source code and can be modified; this is important for advanced applications. Other terminal packages, such as Radio Shack's *Videotex*, should work just fine for a demonstration, but you will need two copies of the program.

Third, you need an extra printer cable. I used Radio Shack's standard cable. The following instructions reference it specifically.

Take a look at your Color Computer operation manual. On Page 26 it shows the configuration of the RS-232 output port. Pin 1 is the carrier detect line; it is used in operating a modem (checks to see if the line is connected). We won't use it for this project. Pin 2 receives data, Pin 3 is the ground wire and Pin 4 transmits data.

Connect the RS-232 ports of two CoCos. Pin 2 of one computer must be connected to Pin 4 of the other. Pin 3 of one is connected to Pin 3 of the other. Pin 4 of one is connected to Pin 2 of the other. This hooks the receive of one to the send of the other and vice versa.

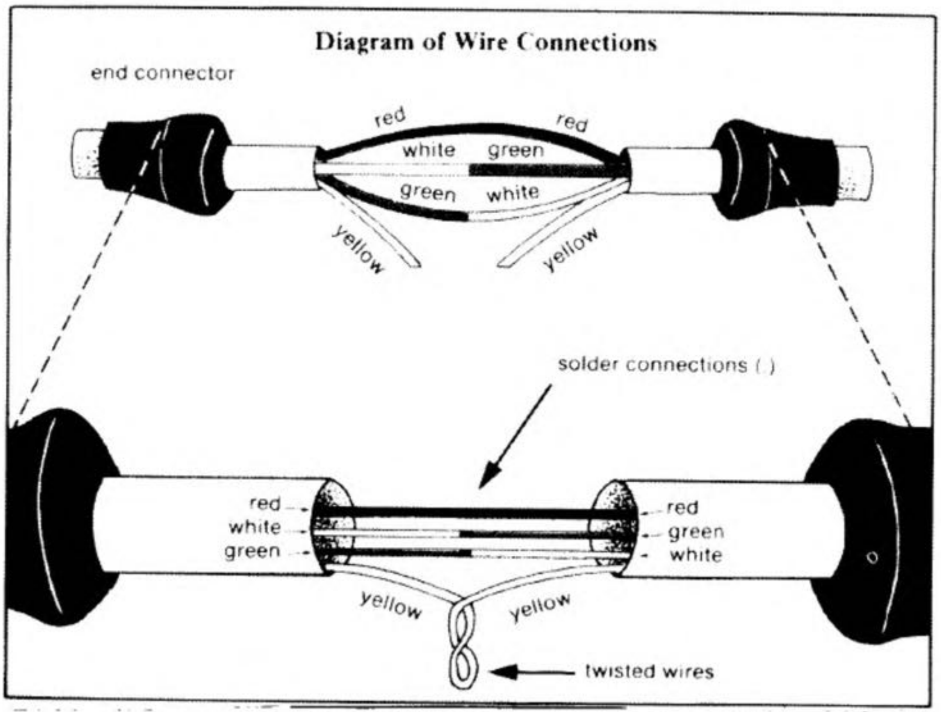
Cut the cable (the extra one you bought) in half. You will see there are four wires inside it. The yellow one goes to the two Pin 1s, green is Pin 2, red is Pin 3 and white is Pin 4.

Rewire the cable this way: Connect the reds back together again. Connect the white of one to the green of the other and the green of one to the other white. The yellow can be left hanging. I just twisted the wires together without soldering them, but be sure to use tape since shorts must be avoided.

Now plug the two ends of the new cable into the printer ports of the two CoCos. Next, load and run the terminal package on both computers at the same time. Once they are running, every time a key is pressed on one computer it will show up on the screen of the other. Data is being sent from one computer to the other.

There are some interesting applications with this experiment. Modify the terminal package to do something with the data it is receiving. You could wrap a game around it. Each player would have their own keyboard and screen and play against each other (you may wish to step up the Baud rate of your package as far as it can reliably go). Another idea might be to double your RAM space for certain application programs. Use your imagination to see what else you can dream up!

Diagram of Wire Connections



Harvesting Summer's Bumper Crop Of BBSs

By R.Wayne Day

This summer turned out to be quite interesting from a telecommunications standpoint, with several intercontinental BBS exchanges going on and just oodles of new "domestic" boards going online.

It all started this July when a caller logged on to TBBS Fort Worth from Sydney, New South Wales, Australia, roughly the farthest point you can get from Fort Worth.

Gerd Oblack, an active OS-9 user and CoCo telecommunicator, wanted to check out the BBS action stateside a bit and let us know what was happening "Down Under." Gerd, who is a member of the New South Wales OS-9 Users Group in Sydney, was able to make a few contacts with the OS-9 users on TBBS.

Graeme Nichols is the SysOp of the NSW OS-9 BBS, which has been operating on Tuesday and Thursday from 7 p.m. to 9 p.m. (Sydney time) for the locals.

I got up real early (3 a.m. Central time) one Monday morning (Australia lies across the International Dateline, so when it's Monday in the United States, it's Tuesday in Australia) and placed the international call. A short while later, after getting the modem standards set up, I was able to spend a little bit of time browsing around the Sydney group's BBS!

That initial success brought about this note from Graeme . . .

"It was great to receive your call from Fort Worth the other night. Just sitting watching my CoCo go through its paces with disk drives whirring and modem lights flashing whilst it talked with your CoCo really drives home just how powerful it is.

"When the other members of the Group were told, they were just as excited and were hopeful that your upcoming article on our BBS will result in some more calls from the U.S.A.

"To this effect, I would like to suggest that 8 a.m. to 12 noon Saturday, local time (in Sydney) would be a convenient time for OS-9 users in the U.S. to call our BBS. I will leave it online with Bell 103 standard, 300 Baud, eight bits, no parity and one stop bit. Unfortunately, my modem cannot handle 1200 Baud FDX (Bell 212 standard), but it can handle CCITT 1200/75 Baud (European standard). That is the standard for our public utility information network, Viatel.

"Please let me know if this is a worthwhile proposition."

By all means, Graeme, I think it's a *smashing* idea (hmmm . . . I may have picked up a bit of the Australian lingo on that call).

So, if you think you would like to invest a couple of dollars in a *real* long-distance call, Graeme's board is available at 61-02-451-2954. Australia's country code for long distance calling is "61" and the city code, much like an area code in North America, is "02."

Is it real expensive to call halfway around the world? A check with AT&T shows that charges for that time of the day run from a high of about $16 to $12 for a 10-minute call, directly dialed. If you're an MCI customer, the appropriate discounts apply to those charges.

During the time when Daylight Savings Time is still in effect in the United States, the times to call are:

Eastern Daylight 6 p.m. - 10 p.m.
Central Daylight 5 p.m. - 9 p.m.
Mountain Daylight 4 p.m. - 8 p.m.
Pacific Daylight 3 p.m. - 7 p.m.

At the same time that North America drops off Daylight Savings time, Australia begins their summer, and Daylight Savings Time, so after November 1, the times are:

Eastern Standard 4 p.m. - 8 p.m.
Central Standard 3 p.m. - 7 p.m.
Mountain Standard 2 p.m. - 6 p.m.
Pacific Standard 1 p.m. - 5 p.m.

Please limit your calls to those times, as the phone is used for other purposes during hours that the BBS is not available.

Software Exchange

One of the byproducts an active BBS'er tends to collect is a good idea of what the "other guy" is doing in regard to home-grown software (the non-commercial type), since most BBSs have a database of various public domain programs. Though calling each and every BBS is the only way to guarantee that you'd have available everything there is to see, a public domain software exchange, coordinated through your local BBS, can be the next best thing.

TBBS Fort Worth and the Sydney OS-9 Users Group have, for example, exchanged copies of our public domain software by the time you read this. Though obviously not all of the software would be applicable (I couldn't bear to fill out an Australian Income Tax form, for example), programming styles and tricks are very worthwhile study materials.

If there's enough interest in doing something organized, perhaps BBS operators could get together, informally, and exchange names and addresses of systems willing to share their public domain programs.

What I envision right now is a mimeographed list of those BBSs wanting to get in on the software exchange. It would be up to the SysOp to make the initial contact with the other system and make arrangements for the exchange.

What do you think? Is there enough interest to start it — and keep it going?

Meanwhile, if you're interested in exchanging public domain programs with the Sydney OS-9 Group, contact:

Graeme Nichols
Sydney OS-9 Users Group
9 Milham Crescent

Forestville, 2087
New South Wales, Australia

And with that, we have the perfect lead-in as we try to answer some common questions from RAINBOW readers.

Questions, Questions . . .

Q: Can I put up programs from THE RAINBOW on my BBS, and what do you mean by "Public Domain"?

A: First, let's define "copyright." A copyright on a program simply means that someone owns the "rights" to a particular creation. When you build a program of your own design, and you do the work, you own a "copyright" to that program; that means unless you let them, no one else can sell, give away or use your program without your permission.

Magazine authors normally assign their rights to a program to the magazine, in all or in part, for payment for the article. In this case, THE RAINBOW purchases the rights to publish and sell that program, so they now own the copyright, in most cases, to all programs that appear in the magazine.

Q: You said "in most cases"?

A: As in all rules, there are exceptions . . . one might look at the *OCNVRT .BAS* program that appeared in this column in the September 1985 issue of THE RAINBOW. It has been released into the public domain, meaning it can be freely distributed without payment to the copyright holder, Mike Ward. Mike gave us permission to include the program in the column, so we could share it with anyone who had a need for it.

The important thing to note, though, is that programs like this are the exceptions to the rule, rather than the norm.

Q: So, public domain means that "all is fair"?

A: Yes, and no.

Public domain is normally meant to say that you can use a particular program for personal use, but you may not sell it or make any charge for using the program. It may be freely distributed, however.

Some authors of public domain programs place restrictions on their programs, though, such as allowing it to be downloaded from one particular BBS or Information Service. Such restrictions are not tacked on frivolously and should be respected.

In general, the rule is that you may post on your BBS (and share on a person-to-person basis) any program that is not copyrighted (like being part of a magazine, unless it's otherwise noted) and does not have any restrictions placed upon it.

Of course, it's always a good practice to give credit where credit is due by making a note of where you got the file and being sure the author's credits remain intact.

Q: "Freeware" is something that's showing up on local BBSs now. What's different about that?

A: Freeware, shareware, etc., is a blending of public domain and copyrighted commercial prorams. Here, the author allows the free distribution of his program, but may either solicit donations based upon whatever the user feels the program is worth (and the author might even suggest a specific amount), or in other cases, the version being distributed is a stripped-down version, which might not do all the things a registered (purchased) version might do.

In any event, software such as this is commercial — it's just that the marketing effort has been redirected from being the traditional package in a store that is advertised in a magazine, to one that is freely distributed among potential customers.

As such, noting any restrictions found in the program or documentation, freeware is normally OK to post on your BBS.

Q: Why this emphasis on public domain programming? Does anyone really care what's in a database on some BBS out in the boondocks?

A: You better believe that a *lot* of folks really care, for two reasons . . .

Putting commercial programs up on a BBS is theft, pure and simple. It's immoral, it's illegal and what's probably worse, in the view of many folks, it fosters the misconception that many computer newcomers and old hands alike may think "it doesn't really hurt anyone."

It *does* hurt, not only the vendor who has lost a sale, but also the operator of the BBS who becomes known as a software thief. It hurts by creating a public misconception of what BBS systems are all about, as well as possibly hurting all BBS users who are beginning to face the threat of legislation that might well regulate some aspects of our hobby.

Secondly, public domain (or public programming as some call it) is probably the purest form of personal computing in existence today.

As Charles Bowen mentioned in *Online Today* in an essay on the subject of public programming, it ". . . promises to make us more self-sufficient consumers with tougher demands on the marketplace for quality, and that's in the truest spirit of personal computing."

Speaking of the Law

The first day of September brought about many changes in the laws of the State of Texas, but what caught most telecommunicators' eye was the newest "Computer Crime" law in the country.

Basically, the new law (Chapter 33, Title 7 of the Texas Penal Code) provides criminal liabilities under two categories of actions: defeating a security system (password protection, for example) or assisting anyone else in defeating such a system (for example, by giving someone a password to a system that he's not supposed to have), and by "harmful access," whereby someone causes a computer to malfunction or "alters, damages or destroys data of a computer program."

Penalties range from a term of up to 180 days in the county jail and/or a $1,000 fine to a prison term of from two to 10 years and/or a fine of up to $5,000.

Much of the local discussion by BBS SysOps around here has centered on the question of whether this new law could be applied to someone who breaks into a privately-run BBS, and the layman's consensus is that there does not appear to be anything in the code that limits prosecutions to the gigantic corporations or commercial data processing establishments. Let's hope we don't have to find out.

Self-Policing of BBSs in Saskatchewan

BBS owners and operators in the Regina, Saskatchewan area of Canada have banded together in an association of SysOps to help keep their area's systems as well run and legitimate as possible. Robert Hamilton, the SysOp of the Phoenix BBS, a CoCo board, reports that the association has had great success, using peer pressure.

If you're in the area, or would like more information on how your area could start a SysOp's Association, contact:

Robert Hamilton
SA SysOp
P.O. Box 401
Regina, SA

New Videotext Services

As the popularity of online communications services has grown over the past few years, there has been an upsurge in local and regional videotext services, as well as nationally advertised speciality networks.

The attorneys of the world have WESTLAW, the Biomedical Community has BMEDSS (Biomedical Engineering Decision Support Services), the physicians have the AMANet, and now, those looking for contacts with "that special someone" have not one, but two online systems devoted to helping you meet that special person.

"You will be meeting someone new. Someone you've never met, yet already know. This is not a futuristic fantasy. It's INTERLUDE . . . "according to an ad from INTERLUDE, a Grand Prairie, Texas, company.

For the introductory price of $75, INTERLUDE provides you with a terminal and a hookup to your television set, along with a local access number to dial to connect into their network.

CVC ONLINE, a New York City information service, has an initial fee of $12.95, and provides an electronic matching service usable with any terminal program . . . "all you need is a personal computer to join in the fun."

Both services are limiting their clientele to adults only, by the way.

For more information on INTERLUDE, you can call (214) 660-2874. For more information on CVC ONLINE, you can contact them at (212) 972-4719 or write: 801 Second Ave., New York, NY 10017.

CCSig Milestone

On one of the more conventional information services, The Color SIG reached a major milestone this summer when the system processed the 100,000th message on CCSig.

In just a little bit over three years online, CCSig has proven to be the longest-running online CoCo information service, processing an average of 90 messages a day over those three years.

So, for all those present and past who have made the CCSig what it is today, a hearty "Thanks!", and hope that the next 100,000 is just as fun!

Mikeyterm Update

There is now a total of three versions of *Mikeyterm* that have been issued by Mike Ward, *Mikey*'s creator.

Version 2.3 is the entry-level version, supporting 300 Baud through the normal RS-232 port on the CoCo, and using 32 by 16 video (normal CoCo text screen).

Version 2.5 requires the use of an RS-232 ROM pack or PBJ-2SP Serial card to provide either 300 or 1200 Baud service using the normal text screen.

Version 3.5, the latest version to be released, also requires the RS-232 ROM pack or PBJ-2SP Serial card, and will support both the normal text screen or the 80-column screen of the PBJ *Word-Pak* (either *Word-Pak I* or *II*) if one is installed. It also supports 300 or 1200 Baud.

A note to users of Version 2.5 or 3.5, as well as other terminal program users that are using the RS-232 ROM pack, you may find that your system appears to be "locked up" unless you provide the RS-232 pack with an apparent "Carrier Detect."

In the case of a Radio Shack Modem II, there is a switch on the back of the modem that allows the carrier to be held high at all times, enabling the sending of auto-dialing instructions, which do work with a CoCo (regardless of what another CoCo magazine reported this spring).

The Hayes Smartmodem provides for a carrier signal to be sent to the modem at all times through the use of a programming switch on the front of the circuit board.

If you're having trouble with any of the terminal programs that use the RS-232 ROM pack, check your modem instructions to see whether you can fool the system into believing there is always a carrier present.

We had planned on including a comparison chart of the various BBS systems available to CoCo owners this month, but circumstances prevent us from living up to that promise at this time. In an effort to be fair to all concerned, the comparison chart is now planned to be included in January's "CommLink."

I hope we had a chance to meet each of you at the RAINBOWfest in Princeton, N.J., in October. It's always enjoyable to get together with RAINBOW readers and swap interesting BBS numbers. If you didn't get a chance to be there, hopefully we'll catch you at the next show!

As always, you can read me several different ways — through the editorial offices of THE RAINBOW, on CompuServe's CCSig (GO COCO) (User ID: 76703,376), through TBBS Fort Worth (817) 232-2087 (300/1200/2400 Baud) or at P.O. Box 79074, Fort Worth, TX 76179. If you write, and would like a direct reply, please be sure to enclose a self-addressed, stamped envelope. □

Apple Users Group ABBS 24 Hours Est.	(02) 451-6575
Ausborne (Osborne) RCPM 24 Hours Est. **1	(02) 95-5377
Club-80 (Sydtrug) RTRS 24 Hours Est.	(02) 332-2494
CoCoLink (Tandy Computers) 24 Hours Est. **10	(075) 32-6370
Date BBS (Computer Dating) Mon & Wed : 1800-2300 Est. Sat : 1200-1700 Est.	(02) 516-3805
Dick Smith RIBM 24 Hours Est.	(02) 887-2276
Info-Center (Paris Radio) 24 Hours Est. **2	(02) 344-9511
Keeboard TBBS Weekdays : 1730-0830 Est. Weekends : 24 Hours Est.	(02) 63103282
Mi Computer Club RCPM 24 Hours Est.	(02) 662-1686
Micro Design Lab. RCPM 24 Hours Est. **3	(02) 663-0151
Omen RTRS Mon-Thurs : 1630-0900 Est. Fri-Mon : 24 Hours Est.	(02) 498-2495
Oracle TBBS Mon-Fri : 0000-1800 Est. Weekends : 24 Hours Est.	(02) 960-3641
Prophet TBBS 24 Hours Est.	(02) 628-7030
Runx Unix System 24 Hours Est. **5	(02) 48-3831
Sorcerer RCPM Weekdays : 1800-0800 (Ring Back) Weekends : 24 Hours Est.	(02) 387-4439
Sydney Public Access RCPM 24 Hours Est. **6	(02) 808-3536
Sydney TRS-80 User Group 24 Hours Est.	(02) 332-2494
Sydney PC User Group 24 Hours Est.	(02) 238-9034
Texas Instruments (TISHUG) Mon-Fri : 1900-0700 Est. Weekends : 24 Hours Est.	(02) 560-0926
North Shore CoCo 1800-2400 Est.(Voice Line) **7	(02) 411-3336
Tandy Access 24 Hours Est.	(02) 625-8071
North Shore Os9 Tues & Thurs : 1930-2130	(02) 451-2954
Sydney Unix System 24 Hours Est.	(02) 487-2533
CoCo Os9 Bulletin Board 2100-0900 Est.	(074) 30-2468
Newcastle Mcc RCPM Weekdays : 1700-0830 Est. Weekends : 24 Hours	(049) 68-5385

Pc Connection BBS Weekdays : 2100-1800 Est. Weekends : 1600-1000 Est.	(03) 528-3750
Omen IV RTRS 24 Hours Est.	(03) 846-4034
Hi Tech 24 Hours Est.	(03) 397-1165
Melbourne Micro Computers CBBS 24 Hours Est.	(03) 762-5088
Sorcerer Computer Users CBBS 24 Hours Est.	(03) 836-4616
Tardis RCMP Weekdays : 1800-0900 Est. Weekends : 24 Hours	(03) 67-7760
Gippsland RCPM 24 Hours Est.	(03) 34-1563
The Australian Beginning 24 Hours Est. **8	(03) 813-3522
CoCo 24 Hours Est.	(03) 397-1165
Telebraille 24 Hours Est.	(03) 755-1341
P.R. Systems BBS Mon-Sun : 2200-0700	(03) 842-6857
Gippsland Mailbus 24 Hours Est.	(051) "27-7245
Competron 24 Hours Est.	(07) 52-9498
Hi-Tech Software 24 Hours Est.	(07) 38-3852
Texas Instruments 24 Hours Est.	(07) 263-6161
Brisbane Tomorrow 24 Hours Est.	(07) 394-2300
Bex 24 Hours Est.	(07) 393-3151
Software Tools RCPM 24 Hours Est.	(07) 378-9530
Rainboard (Tandy Computers) 24 Hours Est. **10	(075) 32-6370
Micsig 24 Hours Est.	(062) 85-1026
Electronic Oracle 24 Hours Est.	(08) 260-6686
Adelaide Micro Users BBS Weekends & Public Holidays only .: 1000:2200 Central Time **8	(08) 271-2043
Computer Ventures BBS 24 Hours Central Time	(08) 255-9146
Outback RCPM 24 Hours Central Time	(089) 27-7111
Omen II RTRS 24 Hours Central Time	(089) 27-4454
Omen III RTRS 24 Hours Western Standard Time	(09) 279-8555
Perth CPM 1800-2100 Western Standard Time	(09) 367-6068
Perth PMP/M 1800-2100 Western Standard Time	(09) 381-6070
Perth CoCo Users Group 24 Hours Est.	(09) 307-1397
The Fountain Texas Instruments 1800-2100 Western Standard Time **9	(09) 272-5931
Launceston 24 hours Est.	(003) 34-0911

So What Do The Stars Mean?

**1 : To gain access type OSBORNE, to read BBS type MINIRBBS
**2 : To logon (ENTER) infocenter, (ENTER) visitor to Phone No. prompt
**3 : To enter system, (ENTER) DDT
**4 : To enter system, (ENTER) TRSDOS
**5 : (02) 48-3831 - Status line
(02) 487-2533 - System line
**6 : (ENTER) 100 for User No. Only one visitor admitted to system per hour, therefore call just after hour to gain acess.
**7 : Call by voice first to gain access to system.
**8 : (ENTER) VISTOR for username & password
**9 : (ENTER) COMPAUST for username.
**10: Vistors, press (ENTER)

CoBBS:
A Look At The Commands

By Richard Duncan

Welcome back! Last month we introduced you to *CoBBS*, a bulletin board program designed to give you, the SysOp, the power you need. This month we will look at the commands available, discuss the logon routine and provide the listings for logon, start-up and creating the RS-232 driver.

Table 1 is a list of the various commands available to the SysOp. When *CoBBS* boots up into the main system it goes out to disk and gets information from a file named *MENU/SYS*. This file contains the data required to set up a menu for the user. A menu can be any number up to 255. A discussion of the menu file will be done later with how to actually set up the menu.

What we need to remember now is that each menu consists of different commands, which are called command types. A command type is an ASCII character. For instance, the command used to download is a Type 'D'. Certain types of commands require additional information and that is contained in the DATA. We will discuss the additional data required when needed.

Load Commands

Types !, #, $, (and) are considered Load commands. These are used to allow access to the various editors (load and run them) from a BBS menu without the need to exit into BASIC. None of the editors have to be on the disk for the system to operate normally — whether they are or not depends on your convenience and disk space. No optional data is required for any of these types.

List Commands

Types 0, 1, 2 and 3 are LIST commands. These are used to list various text files on the disk. Type 0 is for HELP files. Additional information is required and the format is BNAME. Optional data can contain up to five characters, which this one does. The 'B' part of it is for "buffer control." If B=0 then the file will just list; if B=1 then the system will send an open buffer control code to the user, list the file and close the buffer. The "NAME" part is a four-character name to specify which file to list.

The BBS creates the name of the file to list from the LIST type and the four-character name. If you specified the optional data to be "1CLUB," then the system takes the name CLUB and adds HELP in front of it since this is a HELP list. It also adds the suffix "/TXT." Since a '1' was specified, the system opens the user's buffer, lists the file *HELPCLUB/TXT* and closes the buffer. You do not need to specify the drive number as the BBS searches all of your drives for the file.

The only difference between the HELP type and the others is the first four letters of the filename. A Type 3 command that has the same information as above in its DATA will search for a file called *BULLCLUB/TXT*.

Leave Commands

There are three leave commands: types L, N and Y. A Type L command is the standard leave. It allows a user to leave a message to another user on the system. The menu number to post the message to is put in the optional data. If a 'P' is included in the data then the user may specify that the message be private, otherwise only public messages are allowed. Putting a 'K' in the data will tell the system to kill the message after it has been received.

Type N is the same as 'L' except this type only allows private messages. It is considered the "Email" type. Type Y is the SysOp message leave. It is posted to menu 255 and is always addressed to the SysOp. No one except those with SysOp privileges will be able to see this message.

You can specify that the message be posted to any menu other than 255. If you specify a menu between 0 and 249, then the message is posted to that number and a Read command is required, which allows viewing of that number. Specifying a number between 250 and 254 causes the system to ask which number the user wants to post that message to. He can specify any menu number that he is authorized on, including 250 through 254. If you specify 255 as the number, then another number must be given by the user and can be any active menu on which he is authorized.

Read Commands

Read commands allow the user access to messages posted to the system. There are four commands types: R, V, W and X. Type R is the standard read. It allows the user to read any message posted to the number specified in its data if that message is a public message. A private message may be read if it is *to* or *from* that user. A Type V is the same except it allows the user to read all messages (public and private) posted to the number specified in DATA, even if not addressed to him.

Types W and X take the privilege one step farther. A Type W allows the user to read all public messages and private messages (if to or from him) posted to *any* number on or below the number specified in DATA. This is a kind of combined message read. The Type X is the same except it allows the reading of all messages, like Type V, posted to the number specified in DATA or below.

Messages may be read in a forward or reverse scan, or individually. There is also a NEW mode in which the system shows the user all the messages that have been posted since the last time he was on. While the message is being read, pressing a 'P' pauses the message scroll and pressing ENTER resumes it. Also, an 'N' stops that message and goes to the next one, while an 'S' stops the Read function and returns to the menu.

If a message is deleted the user is given the opportunity to reply. Make special note of one thing: The system determines who is authorized to reply to a message by the privilege of the *first* Leave command on the menu he is using. If a privilege of 30 is required to leave a message and the user has a privilege of 25, he may not reply to that message even if it is addressed to him.

A menu may contain several different types of Leave commands, but the first one of the list is the one that determines the required privilege to reply to a message.

There are four additional Read commands. Types S, 7, 8 and 9 are SCAN types. These operate the same as their Read counterparts, except only the header of the message is printed and the user is then given the option of reading the text or continuing.

Download Commands

This is the standard download of the board, and all files must be in ASCII. A good understanding is required of the way the system determines the filenames.

The download type has a privilege level and 16 flags as do all commands of *CoBBS*. The DATA is formatted as follows: BNAME, where 'B' is the buffer code and "NAME" is the download's four-letter name.

The best way to understand how the downloads function is to take an example. Consider the following command specification.

```
Type: D
DATA: 0COCO
```

DATA is "0COCO." The buffer number is zero; this means when the directory of programs available is listed the user's buffer will not be opened and closed automatically. A '1' as a buffer number would specify "open buffer, list directory, close buffer."

The download's name is "COCO." The system takes the name COCO and adds the following: DOWN + name + /MNU so the name of the text that contains the directory of programs available is *DOWNCOCO/MNU*. The directory should contain a program number or designation for each item listed. An example of a download directory is:

1) *INVENT/BAS* — Program to maintain your inventory.

2) *DIRPRINT/BAS* — Print out your disk directory.

3) *STREK/BAS* — Excitement of Captain Kirk.

The user is then prompted for the number to download. This is why you want to set up a directory preferably with numbers. The download command takes this number to specify the file to be downloaded. The format for this is NAME + number + /DOW. For our example, if the user chooses number '1', then the file the system looks for is *COCO1/DOW*.

There is a totally different type of download available and it is set up by using a '3' or '4' in the buffer code of the DATA statement. While a '0' or '1' determines whether or not the user's buffer is opened when the menu is listed, it takes on a different meaning when the code number is greater than two. The format for the DATA statement in these cases is: BDPxx, where 'B' = download type; 'D' = drive to access for downloads; 'P' — if a 'P' is included then the user may not download any program with an extension of "/SYS," "/BAK" or "/BIN."

When a '3' or '4' is specified in DATA the user is prompted with "FILE TO DOWNLOAD:" only. He supplies the filename desired. If a '3' is used the system limits the user's access to only the drive specified in DATA. If a 'P' is included the user may not download any program with extensions noted in the format above. A "B=4" allows full access to all drives. Information that is specified in DATA is ignored. A 'P' restriction may also be used in this type.

One other comma.id is available when using a '3' or '4'. When prompted for the file to download, a directory may be obtained and listed by using the DIR command for the drive desired.

Upload Commands

All uploads are the same for any menu, but there are options available to the SysOp determined by DATA. The format of DATA is: TDxxx, where, T = type of upload; D = drive to upload to; xxx = ignored by system.

When T=1 the system uploads to the specified drive in DATA. The user provides an eight-character filename and the system adds a "/UPL" extension. If a file exists already it may not be overwritten. If T=2 the user specifies the filename and extension. Upload goes to the drive specified in DATA. No overwrites are permitted. A T=3 allows the user to specify filename, extension and drive for the file to write to, but will not allow a file to be overwritten. A T=4 lets the user specify the filename, extension and drive for the file to write to. If the file exists, it will be overwritten.

Menu Control File

Type @ is a menu control command for the user. This command gives the user some control in how the menus list and which menu he will access the system when logging on. The first part is the Menu Control Editor. This allows the user to specify which menu he will initially use after getting through the logon routine. It is normally set to menu zero through *SCF/EDI* for new users, but may be changed by the user to any menu that he is authorized. If changed, the next time the user calls he automatically starts on that menu after logging in.

The second part of this command allows the user to determine how the various menus will prompt him from full menu listing to just seeing "COMMAND" only. Even if a user chooses a higher menu prompting, if he gets lost or cannot remember what the various commands are, he may hit the carriage return key to get a full listing of the menu.

System Movement

The following commands are used to move between menus and also out of the system.

Type A is considered an ABORT command. This will exit a menu and go to another one based on DATA. The following types are available:

0 — Return to menu the user last came from
1 — Return to menu specified as the logon menu
2 — Return to menu zero

Type E to exit and run the BASIC program specified in DATA. (Note: If no extension is given on a filename it is assumed to be "/BAS."

Type G is a GOTO command. The system will "goto" the menu number specified in DATA.

Type Q is the same as END. It stops the board program and puts the user into BASIC.

DATA: 0 — Exit immediately
1 — Exit with password

The format of DATA is TPASS, where T=type of quit; PASS=four-character password if Type 1.

Type T terminates the user's session. DATA specifies whether the user is asked if he wants to leave a SysOp message or not. DATA types are:

0 — Terminate immediately
1 — Allow the user to leave a private message to the SysOp

Type Z causes the system to restart as if the user has first called.

Miscellaneous Commands

Type * command shows the user start time, end time and time on system.

Type C is a "chat" command. It pages the system with audio tones, then prompts the user to continue using the system until the SysOp breaks in. Entry into chat is by pressing the space bar. To exit chat mode, press SHIFT and CLEAR, then ENTER on a line by itself.

Type O lists the user in the Userlog file (*USERL/SYS*). It either searches for a certain user or lists the whole file. New users will not be displayed. This way, a nuisance caller cannot access the system and see his "name" up in lights. It displays Probationary users and Registered users.

Type P is a password change command. It allows the user to change the password that he uses to access the system.

Logon Routine

After going through a BOOT process the system comes up with a screen that has the BBS's name, copyright, message numbers, etc. The system waits until it either receives a carrier or gets one of the SysOp's key commands.

Upon receiving a carrier, the system prints the board's name and copyright. (I feel like it is very little to ask that this be retained. Since this is one of the few requirements I had when releasing it for distribution, I hope that you would be decent enough to give me credit for my time and work.)

After the copyright is displayed, the system checks to see if it has received a Control-C or an '*'. If so, before it lists a *SIGNON/TXT*, it prints the prompt PLEASE PRESS <ENTER> TO CONTINUE:. At this point, you may enter a password and exit directly into BASIC. This password is specified in Line 22 of *USER/SYS* and should be changed to whatever you want. Since it is a string the password may be up to 250 characters long. If no Control-C is received, the system goes on and lists *SIGNON/TXT*.

The system then asks for a user number. This feature allows a user to have quicker access. The user number is actually the record number of his entry and is given to the Registered users after he logs on and sees the system information. Note: The use of this type access requires that users must not be deleted from the userlog or all numbers past that point will be changed. If this type of access is not desired, then remove Line 68 from *USER/SYS*.

Users who do not have a number, or if it is not used, are requested to enter their name and where they are from. They may use a semi-colon to separate "name;from" to enter it all on one line. If there is no space in the name they are requested to enter it again, thus, hopefully requiring them to enter first and last name. If the information they entered is more than 90 characters long, then they are dropped. This is helpful for the nuisance calls. The system then checks the *USERL/SYS* file for any record of them. If found, it will request their password and give access if entered correctly.

There are four different ways you can specify that a caller be handled when logging in:

1) If not already in the Userlog, the system disconnects.
2) User is requested registration information and then terminated.
3) Same as '2' but the system displays an ASCII text file, *NEWUSER/TXT*, and disconnects.
4) Normal operation. New user will register and access.

Of course, a caller who is in the Userlog will access the system normally.

After all the above has been completed, the user will be into the system. If specified by *SCF/EDI* (System Control File, to be discussed later) the system checks for any messages to the user. This routine may be aborted by the end user by pressing the 'S' key during the check. If a message is waiting, the user is shown its number, menu name, who it is from and the subject.

Access is now complete and the main BBS routine is loaded and run.

Starting Up

Some of the routines required to boot up the system will be presented in Part 3 of this series, which includes the System Control File Editor (*SCF/EDI*) and the editors to create the file *MENU/SYS*. Let's look though what is involved in booting up the BBS.

1) Format a disk for each drive to be used on the system. The one referred to as the "system disk" is the one in Drive 0.

2) Load and run the System Control File editor and create the system parameters you desire to match your equipment online.

3) There are three programs that are only used once when booting up the system. If you do not wish to keep them online and save disk space, format another disk for these files and copy them over. These are *64/BAS*, *STARTUP/BAS* and *COTERM/SYS*.

4) Copy the following files over to the system disk that will be in Drive 0: *USER/SYS* and *COBBS/SYS*.

5) Copy the following files to the disk that will keep that file as specified by *SCF*: *MENU/SYS*, *USERL/SYS*, *HDR/SYS* and *MSG/SYS*. For example, if you specified the message header to be on Drive 1, then copy *HDR/SYS* to the disk that will be in Drive 1.

6) If you are going to call up an editor from the board then copy over the ones desired to Drive 0.

7) Get the system into 64K. Type `PCLEAR1` and press ENTER. *CoBBS* must be run in 64K mode.

8) Load and run *STARTUP/BAS*. Enter the correct date, day (first three letters) and time. You are prompted as to whether or not you want to "PROTECT?" the system. If you answer "yes," the system's BASIC is modified. When under protection, if the board ever breaks into BASIC it locks up. The routine is crude but effective and secure.

You are asked if you want the "C/R Mod?" This modifies the RS-232 driver so the system puts out a `CHR$(13)` when no carrier is being detected.

This mod requires a few additional steps for the SysOp to use the system locally. Before exiting the system, or using the SysOp logon, it is necessary to remove the modification. This is done within the system by using its exiting commands. To logon locally, you must first exit into BASIC with the SHIFT and up-arrow key, then type `RUN`. When the screen reappears, press the '@' key as you normally would. To reinstall the modification in either *USER/SYS* or *COBBS/SYS*, type `GOTO10000` from BASIC. Typing this in *USER/SYS* completely reboots and reprotects the system, including dropping the modem. Doing it while *COBBS/SYS* is loaded reprotects and starts *COBBS/SYS* again, but doesn't drop the modem.

If the boot programs are put on a separate disk, insert the system disk in Drive 0 before answering the protect question. After answering, the system loads the logon routine and presents you with the standby screen.

With the system protected, the SysOp may not break into BASIC with the BREAK key, but presses and holds the SHIFT key. Next, press the up-arrow key while still holding down the SHIFT key. Once you break into BASIC this way the system is no longer protected. To reprotect, type `GOTO10000` while the logon routine is in memory.

Conclusion

We will continue next month and look at all the editors required to get the system up and maintain the Userlog. I will be glad to provide you with a disk of all the programs and also a disk with expanded documentation for $25. If you have any questions you can call my BBS at (501) 735-5614 online 24 hours a day, or write to me at 2504 N. Gathings Drive, West Memphis, AR 72301. □

Table 1

Command Types

Type	Name	Data
!	LOAD "SMF/EDI",R	
"	Reserved	
#	LOAD "SUL/EDI",R	
$	LOAD "SCF/EDI",R	
%	Reserved	
&	Reserved	
'	Print messages	0 or 1
(	LOAD "LOOKTRAC/BAS",R	
)	LOAD "LOOKREG/BAS",R	
*	Time on system	
+	SysOp option	
,	SysOp option	
-	SysOp option	
.	SysOp option	
/	SysOp option	
0	List HELP file	BNAME
1	List INFO file	BNAME
2	List TEXT file	BNAME
3	List BULL file	BNAME
4	Reserved	
5	Reserved	
6	Reserved	
7	Scan, 1 menu, all messages	Menu #
8	Scan, menu down, private TO/FROM	Menu #
9	Scan, menu down, all messages	Menu #
:	Reserved	
;	Reserved	
<	Reserved	
=	Reserved	
>	Reserved	
?	Reserved	
@	Menu control editor	
A	Abort out of menu	0,1,2
B	Reserved	
C	Chat. Page system	
D	Download a file	BNAME
E	Exit and run program	program name
F	Load and run file specified	
G	Goto menu #	Menu #
H	Reserved	
I	Reserved	
J	Reserved	
K	Kill messages	
L	Leave messages (standard)	Menu #, (P)
M	Read E-mail (private)	Menu #
N	Leave E-mail (private)	Menu #
O	List Userlog	
P	Password enter/change	
Q	Quit. Exit to BASIC	0,1
R	Read messages (standard)	Menu #
S	Scan messages (standard)	Menu #
T	Terminate	0,1
U	Upload an ASCII file	0,1,2,3,4
V	Read, 1 menu, all messages	Menu #
W	Read, menu down, TO/FROM	Menu #
X	Read, menu down, all	Menu #
Y	Leave SysOp message	
Z	Restart logon	

220	220
570	28
750	203
END	227

Listing 1: STARTUP

```
5 CLEAR1ØØØ
1Ø POKE65386,1Ø6:CLS
2Ø PRINT"-COBBS BBS STARTUP-"
3Ø PRINT"-         BY          -"
4Ø PRINT"-RICHARD  DUNCAN   -"
5Ø PRINT"-    <C> 1985        -"
6Ø PRINT:PRINT
7Ø LOADM"COTERM"
8Ø DEFUSRØ=&HØEDØ
9Ø DEFUSR1=&HØED3
1ØØ POKE65387,54
11Ø POKE65386,1Ø6
12Ø PRINT"PNPCLK/BAS - <C> A. B.
 TREVOR"
13Ø INPUT"DATE (MM,DD,YY)";MM,DD
,YY
14Ø POKE46Ø8,MM:POKE46Ø9,DD:POKE
461Ø,YY
15Ø LINEINPUT"DAY: ";A$
16Ø K=(INSTR(1,"SUMOTUWETHFRSA",
LEFT$(A$,2))-1)/2:X=USRØ(K)
17Ø INPUT"TIME (HH:MM)";HR,MN
18Ø X=USR1(HR*256+MN)
19Ø X=VAL(HEX$(PEEK(&HØEFD))):PO
KE4611,X
2ØØ EXEC&H1Ø91:PRINT"AT SØ=1 EØ
S2=255":POKE&H11ØA,&H12:POKE&H11
ØB,&H12:POKE&H113C,&H12:POKE&H11
3D,&H12:PRINT:PRINT
21Ø FOR A=&H1ØD5 TO &H1ØD7:POKE
A,&H12:NEXTA:GOTO25Ø
22Ø PRINT"LOADING LOG-ON ROUTINE
"
23Ø GOSUB6ØØØ:GOTO59Ø
24Ø PCLEAR1:GOTO7Ø
25Ø PRINT:PRINT"ERROR TRAP - BY
ANDY KLUCK
26Ø CLEAR2ØØ:GOSUB5ØØ
27Ø AD=&HØE1Ø
28Ø EX=AD
29Ø FOR AD=AD TO AD+&H7B
3ØØ READD$
31Ø POKEAD,VAL("&H"+D$)
32Ø NEXT
33Ø EXEC EX
34Ø GOTO22Ø
5ØØ DEFFNPL(X)=PEEK(X)*&H1ØØ+PEE
K(X+1):RETURN
51Ø DATA CC,7E,39,E7,8C,FA,3Ø,8D
,Ø,15,F6,1,94,FE,1,95
52Ø DATA E7,8D,Ø,1E,EF,8D,Ø,1B,B
```

```
7,1,94,BF,1,95,39,1F
53Ø DATA A9,35,4Ø,81,54,27,1Ø,34
,4Ø,34,4,C6,39,F7,1,8E
54Ø DATA 35,1,7E,FF,FF,FF,FF,9D,
9F,BD,AF,67,C6,39,F7,1
55Ø DATA 8E,DC,2B,27,15,BD,AD,1,
25,11,AF,8D,FF,E7,3Ø,8D
56Ø DATA Ø,C,BF,1,8F,C6,7E,F7,1,
8E,39,7E,AE,D2,86,39
57Ø DATA B7,1,8E,9E,68,3Ø,1,27,F
1,D7,FD,DC,68,DD,FE,BD
58Ø DATA AD,33,AE,8D,FF,BF,BD,AE
,BB,7E,AD,9E
59Ø PRINT"INSERT SYSTEM DISK."
6ØØ LINEINPUT"PROTECT? ";A$:IFLE
FT$(A$,1)="Y" AND PEEK(4684)<>18
8 THENGOSUB64Ø
61Ø GOTO71Ø
62Ø LINEINPUT"C/R MOD? ";A$:IFLE
FT$(A$,1)="Y"THEN1ØØØØ
625 LOAD"USER/SYS",R
63Ø END
64Ø L=&HØEØØ
65Ø POKE&HAC7D,&HØE:POKE&HAC7E,&
HØØ
66Ø POKEL,&H86:POKEL+1,Ø
67Ø POKEL+2,&HB7:POKEL+3,&HFF:PO
KEL+4,&H4Ø
68Ø POKEL+5,&HB7:POKEL+6,&HFF:PO
KEL+7,&H6A
69Ø POKEL+8,&H7E:POKEL+9,&HA3:PO
KEL+1Ø,&H9Ø
7ØØ RETURN
71Ø '-SYSTEM CONTROL FILE-
715 CLEAR1ØØØ
72Ø DSKI$ Ø,17,18,S1$,S2$
73Ø POKE4629,ASC(MID$(S1$,9,1))
74Ø POKE4664,ASC(MID$(S1$,1Ø,1))
75Ø POKE4663,ASC(MID$(S1$,11,1))
76Ø FOR A=Ø TO 2:POKE466Ø+A,ASC(
MID$(S1$,12+A,1)):NEXT A
77Ø POKE4628,ASC(MID$(S1$,15,1))
78Ø POKE4666,ASC(MID$(S1$,16,1))
79Ø POKE4669,ASC(MID$(S1$,17,1))
8ØØ POKE4679,ASC(MID$(S1$,18,1))
81Ø POKE468Ø,ASC(MID$(S1$,19,1))
82Ø POKE4667,ASC(MID$(S1$,2Ø,1))
83Ø POKE4659,ASC(MID$(S1$,21,1))
84Ø POKE4694,ASC(MID$(S1$,22,1))
85Ø FORA=ØTO6:POKE467Ø+A,ASC(MID
$(S1$,23+A,1)):NEXT A
86Ø GOTO62Ø
6ØØØ PRINT:PRINT"DISK DOS TYPE"
6ØØ5 PRINT"  A - DOS 1.Ø":PRINT"
  B - DOS 1.1"
6Ø1Ø LINEINPUT" >";A$
6Ø15 IFA$="A"THEN6Ø3Ø
6Ø2Ø IFA$="B"THEN6Ø5Ø
6Ø25 GOTO6ØØØ
6Ø3Ø POKE4681,&HCB:POKE4682,&H4A
6Ø35 POKE4683,197:POKE4684,143
6Ø4Ø RETURN
6Ø5Ø POKE4681,&HCC:POKE4682,&H1C
6Ø55 POKE4683,197:POKE4684,188
6Ø6Ø RETURN
1ØØØØ REM
1ØØØ5 X=&H112B:POKE X,&H86:POKE
X+1,&HØD:POKE X+2,&HB7:POKE X+3,
&H11:POKE X+4,&H81:POKE &H1127,&
H1Ø
1ØØ1Ø GOTO625
```

37	218	740	178
70	174	940	239
110	188	4125	201
175	21	7055	8
211	75	8075	189
240	149	8120	162
268	234	9115	174
300	22	9440	227
371	229	9710	184
455	24	END	59
535	11		

Listing 2: USER SYS

```
Ø '-COBBS (USER/SYS) <C>1985
     RICHARD DUNCAN (7/8/85)
5 '-LOGIN ROUTINE-
1Ø CLEAR5ØØ:RUNT9ØØ:GOSUB81ØØ
15 POKE65386,1Ø7:CLOSE:UNLOAD
2Ø CLEAR4ØØØ:DIM BN$(255):NU$=CH
R$(Ø):RUNT9ØØ:DEFUSRØ=&HØED3:CJ$
=CHR$(13)+CHR$(8):DR$(Ø)="Ø":DR$
(1)="1":DR$(2)="2":DR$(3)="3":PO
KE4658,255
22 PW$="EXIT PASSWORD"
24 POKE&HØ16B,&H1Ø:POKE&HØ16C,&H
CD:GOSUB82ØØ
25 CLS:PRINT:PRINT:Z=PEEK(65385)
:Z=Z AND 32:IF Z=Ø THEN 4Ø
26 GOSUB8ØØØ:TIMER=Ø
3Ø II$=INKEY$
31 IF II$="@"THENCLS:POKE4658,Ø:
GOTO66ELSEIFII$="_"THEN FOR X=&H
112B TO &H1136:POKE X,&H12:NEXT
X:IFPEEK(4684)<>188THENPOKE&HAC7
D,&HA3:POKE&HAC7E,&H9Ø:ENDELSEEN
D
34 II=TIMER:IFII>19999ANDPEEK(46
94)<>ØTHEN CLSØ:IFII>6ØØØØTHEN T
IMER=2ØØØØ
35 IF II$="*"THENGOSUB8ØØ:GOTO26
36 IF II$<>""THEN4Ø
37 CD=PEEK(65385)AND32:IF CD=ØTH
EN FOR A=1TO2ØØ:NEXT A:GOTO4Ø
38 GOSUB94ØØ
39 GOTO3Ø
4Ø CLS:EXEC&H1ØDA:POKE4657,Ø
45 REM A=JOYSTK(Ø):SP=JOYSTK(2):
IF SP<5 THENPOKE65387,56ELSEPOKE
65387,54
5Ø FOR A=1 TO 1Ø:PRINT:NEXT
6Ø CLS:PRINT"COLOR BBS SYSTEM":P
RINT"CoBBS VERSION 1.2":PRINT"<C
>1984 by Richard Duncan":GOSUB97
ØØ:PRINT:PRINT"CoBBS #xx":PRINT:
PRINT:PRINT:GOSUB12ØØ
61 RUNT65:D=Ø:POKE&HØ16B,197:POK
E&HØ16C,PEEK(4684)
62 F$="SIGNON/TXT:"+DR$(D)
63 OPEN"I",#1,F$
64 RUNT66:IFNOTEOF(1)THENGOSUB97
ØØ:LINEINPUT#1,A$:PRINTA$:GOTO64
ELSECLOSE:D=5
65 D=D+1:IFD<4THEN62
66 RUN T 9ØØ:GOSUB9ØØØ:IFK1=ØTHE
N67ELSEGET#1,1:RE=CVN(H1$):GET#1
,2:RS=CVN(H1$):CLOSE
67 CLOSE:GOSUB94ØØ:POKE4612,HR:P
OKE4613,MN:POKE4614,S:DT$=CHR$(P
EEK(46Ø8))+CHR$(PEEK(46Ø9))+CHR$
(PEEK(461Ø))+CHR$(HR)+CHR$(MN):P
OKE&HØ16B,16:POKE&HØ16C,2Ø5:TR=Ø
68 RUNT981Ø:EXEC&H1ØDA:TR=TR+1:I
F TR>3THEN981ØELSEPRINT:PRINT"US
ER # (Ø=NEW):";:GOSUB675:R5=VAL(
CH$):CH=R5:IF R5>1THEN11ØELSE7Ø
7Ø EXEC&H1ØDA:TR=TR+1:IF TR>3 TH
EN981ØELSEGOSUB97ØØ:RUN T 7Ø:CLO
SE:PRINT:PRINT:PRINT"ENTER FIRST
 AND LAST NAME":PRINT"NAME: ";:G
OSUB675:K=INSTR(CH$,CHR$(32)):IF
K=ØORCH$=""THEN7Ø
71 NA$=CH$:IFLEN(NA$)>9ØTHENPOKE
65387,1Ø6:RUN:ELSEIFLEN(NA$)<5TH
EN7Ø
75 K=INSTR(CH$,";"):IFK>ØTHEN NA
$=LEFT$(CH$,K-1):CH$=RIGHT$(CH$,
LEN(CH$)-K):K=INSTR(CH$,";"):IFK
=ØTHEN LO$=CH$:GOTO9Ø ELSE LO$=L
EFT$(CH$,K-1):GOTO9Ø
8Ø GOSUB97ØØ:RUN T 8Ø:PRINT"FROM
: ";:GOSUB 675:IFCH$=""THEN7ØELS
E LO$=CH$
9Ø PRINT:PRINT:PRINT:PRINTTAB(5)
"NAME: ";NA$:PRINTTAB(5)"FROM: "
;LO$:PRINT:PRINTTAB(5)"CORRECT (
Y/N) ";
95 GOSUB625
1ØØ IFCH$="N"THENPRINT"NO":GOTO7
Ø
1Ø5 IFCH$<>"Y"THEN95ELSE TR$=NA$
+CHR$(13)+LO$+CHR$(13):PRINT"YES
"
11Ø RUN T 9ØØ:PRINT:GOSUB92ØØ:GE
T#1,1:C1=ASC(U7$):C2=ASC(U8$):CN
=C1*256+C2:PRINT:PRINT"CHECKING
USERLOG...";:IF R5>1THEN R=R5:IF
 R>LOF(1)THEN67ELSEGET#1,R:R6=R:
GOTO173
12Ø FOR R=2 TO LOF(1):GOSUB97ØØ
13Ø GET#1,R:R6=R:K=INSTR(UN$,NU$
):IF LEFT$(UN$,K-1)<>NA$ THEN 17
Ø
14Ø L=INSTR(K+1,UN$,NU$):F$=MID$
(UN$,K+1,L-1-K):IF F$=LO$ THEN 1
8Ø
15Ø PRINT:PRINT:PRINTLEFT$(UN$,K
-1):PRINT F$:PRINT"IS THIS YOU?
";:T=Ø:EXEC4314
16Ø GOSUB 625:IF CH$="Y" THEN PR
INT"YES":GOTO18Ø
165 IFCH$<>"N"THEN16ØELSEPRINT:P
RINT"USER NAME EXIST.  PLEASE EN
TER":PRINT"YOUR NAME A LITTLE DI
FFERENT":GOTO7Ø
17Ø NEXT R:PRINT:IFPEEK(468Ø)>ØT
HENPRINT"WRITING USER...":GOTO46
ØELSEPRINT"USER NOT FOUND":GOTO3
ØØ
173 IF UP$<>"        " THEN 18ØE
LSEPRINT"NO PASSWORD IN LOG.":PR
INT"PLEASE ENTER A PASSWORD OF U
P":PRINT"TO 8 CHARACTERS: ";
175 GOSUB655:IF CH$="" OR LEN(CH
$)>8 THEN PRINT"ENTER A PASSWORD
: ";:GOTO175
177 LSET UP$=CH$:GOTO21Ø
18Ø IF UY=1 THEN 21Ø ELSE PRINT:
EXEC&H1ØDA:PRINT"PASSWORD: ";:A$
="":RUN T 18Ø
19Ø EXEC4314:CH$=CHR$(PEEK(4481)
):IC$=INKEY$:IFIC$<>""THEN CH$=I
C$:GOTO191ELSEIFCH$=CHR$(Ø)THEN1
9Ø
191 K=INSTR(CJ$,CH$):ON K+1 GOTO
192,2ØØ,193:GOTO19Ø
192 A$=A$+CH$:IFPEEK(4679)>ØTHEN
PRINTCH$;:GOTO19ØELSEPRINT"*";:G
OTO19Ø
193 IFLEN(A$)>ØTHENA$=LEFT$(A$,L
EN(A$)-1):PRINTCH$;
194 GOTO19Ø
2ØØ R5=RØ:PRINT:A$=LEFT$(A$+STRI
NG$(8,Ø),8):IF A$=UP$ THEN 21Ø E
LSE T=T+1:TR$=TR$+CHR$(13)+A$:IF
 T>2 THEN 98ØØ ELSE18Ø
21Ø RUN T 9ØØ:TR$="":KK=INSTR(UN
$,NU$):NA$=LEFT$(UN$,KK-1):KL=IN
STR(KK+1,UN$,NU$):LO$=MID$(UN$,K
K+1,KL-1-KK):TR$=TR$+NA$+CHR$(13
)+LO$+CHR$(13)+A$:C2=C2+1:IFC2>2
55THEN C2=Ø:C1=C1+1:IFC1>255THEN
 C1=Ø:C2=C2+1
211 LSET U7$=CHR$(C1):LSET U8$=C
HR$(C2):CN=C1*256+C2:GOSUB97ØØ:I
F UY=1 THEN LSET UM$=CHR$(Ø)
215 PUT#1,1:FORA=1TO1Ø:PRINT:NEX
TA:PRINT"WELCOME ";NA$:PRINT"PLE
ASE WAIT....":GOSUB94ØØ:PRINT:PR
INT
22Ø PR=ASC(UA$):F$=U1$:GOSUB41ØØ
:P1$=F$:F$=U2$:GOSUB41ØØ:C3=Ø:P2
$=F$:BC=ASC(UE$):LG=CVN(U5$):LM=
CVN(UM$):DB$=UD$:TI=ASC(UO$):UL=
VAL(UL$):LSET UD$=DT$:POKE 4618,
ASC(UE$):POKE4619,ASC(UO$)
23Ø LSET U7$=CHR$(C1):LSET U8$=C
HR$(C2):F$=UR$:GOSUB41ØØ:R$=F$:L
G=LG+1:LSET U5$=MKN$(LG):LSET UM
$=MKN$(RE):IF UY=1 THEN LG=1
234 IFASC(UR$)>ØTHENPUT#1,R
235 PRINT"LOW SYSTEM MESSAGE";RS
:PRINT"HIGH SYSTEM MESSAGE";RE:G
OSUB97ØØ
236 FORA=1TO5:A(A)=ASC(MID$(DB$,
A,1)):NEXTA:C4=1:PRINT"LAST ON:"
;STR$(A(1));"/";RIGHT$(STR$(A(2)
),2);"/";RIGHT$(STR$(A(3)),2);"
";STR$(A(4));":";:A$=RIGHT$(STR$
(A(5)),2):IFVAL(A$)>9 THEN PRINT
A$ELSEMID$(A$,1,1)="Ø":PRINTA$
237 CN=C1*256+C2:PRINT"YOU ARE C
ALLER NUMBER";CN
24Ø PRINT"YOU HAVE LOGGED IN:";L
G" TIMES":PRINT"LAST MESSAGE REC
EIVED:";LM:PRINT:PRINT"YOU ARE A
UTHORIZED";ASC(UO$)*5;"MINUTES"
245 MW=Ø:IFLOF(1)<2THENPUT#1,2
25Ø CLOSE:PRINT"AS A ";:IFRIGHT$
(R$,1)="Ø"THENPRINT"NEW USER.":P
RINT:GOTO7ØØØ
255 IFMID$(R$,2,1)="1"THENPRINT"
REGISTERED USER #";R ELSEPRINT"P
ROBATIONARY USER."
259 PRINT:IFPEEK(4629)<4THEN98ØØ
ELSEIFPEEK(4659)=ØTHEN UT=1:GOTO
7Ø1Ø
26Ø PRINT:PRINT:IFPEEK(4666)=ØTH
EN292ELSEPRINT"CHECKING FOR MESS
AGES TO YOU.":CS$=NA$+NU$:GOSUB9
7ØØ:CLOSE:GOSUB9ØØØ:F=Ø:IFK1=ØTH
ENCLOSE:GOTO293
265 FOR A=2 TO K1:GOSUB97ØØ
266 EXEC&H1ØDA:CH=PEEK(4481):IFC
H=83ORCH=115THENPRINT"MESSAGE CH
ECK ABORTED.":GOTO292
267 GET#1,A:K=INSTR(H5$,CS$):IFK
=ØTHEN291ELSEK=INSTR(H5$,CHR$(Ø)
):L=INSTR(K+1,H5$,CHR$(Ø)):M=INS
TR(L+1,H5$,CHR$(Ø)):XX=L-K-1:IFX
X<1THEN N$=""GOTO268ELSEN$=MID$(
H5$,K+1,XX)
268 FR$=LEFT$(H5$,K-1):XX=M-L-1:
IFXX<1THENS$=""ELSES$=MID$(H5$,L
+1,XX)
269 IFN$<>NA$ THEN 291
27Ø '
271 M5$=STR$(ASC(H8$))
283 F$=H2$:GOSUB41ØØ:RF$=F$:IFMI
D$(RF$,2,1)="1"ORMID$(RF$,3,1)="
1"THEN291
285 IFF3=ØTHENPRINT:PRINT"MESSAG
ES WAITING!":F3=1:PRINT:MW=1
287 IFMID$(RF$,1,1)="1"THENPRINT
"PRIVATE MESSAGE"
289 PRINT"#";CVN(H1$);" - ";:H8=
ASC(H8$):IFLEN(BN$(H8))=ØTHENPRI
NT"MENU #";M5$ ELSE PRINTBN$(H8)
;" -"
29Ø PRINT"FROM: ";FR$:PRINT"SUBJ
: ";S$:PRINT
291 NEXTA:IF MW=Ø THEN PRINT"SOR
RY, NO MESSAGES FOR YOU.":CLOSE:
GOSUB97ØØ:PRINT
292 CLOSE:GOSUB97ØØ:PRINT
293 TR$=CHR$(13)+STRING$(2Ø,"=")
+CHR$(13)+STR$(CN)+"  "+DA$+" "+
TI$+CHR$(13)+NA$+" / "+LO$+CHR$(
13):GOSUB96ØØ:CLOSE
294 R1=INT(R6/256):R2=R6-(R1*256
):POKE469Ø,R1:POKE4691,R2
295 GOSUB94ØØ:POKE4615,HR:POKE46
16,MN:POKE4617,S:PRINT:LOAD"COBB
S/SYS",R
3ØØ '-NEW USER-
3Ø5 RUN T 9ØØ:IFPEEK(4629)<2THEN
98ØØ
31Ø UY=Ø:'GOTO5ØØØØ
315 PRINT:PRINT:PRINT:PRINT"SORF
Y, YOU ARE NOT ENTERED":PRINT"IN
 THE USER LOG."
32Ø '
335 PRINT:PRINT:PRINT"PLEASE LEA
VE THE FOLLOWING":PRINT"INFORMAT
ION.  ALL QUESTIONS":PRINT"MUST
BE ANSWERED TO BE REGISTERED"
34Ø PRINT:PRINT:GOSUB97ØØ
345 PRINT"NAME: ";NA$:EXEC&H1ØDA
35Ø GOSUB97ØØ:PRINT"ADDRESS: ";:
RUNT35Ø
351 GOSUB65Ø:IFCH$=""THEN35ØELSE
 AD$=CH$
355 PRINT"CITY: ";LO$
36Ø GOSUB97ØØ:PRINT" ZIP: ";:GOS
UB65Ø:IFCH$=""THEN36ØELSE ZP$=CH
$
365 PRINT:PRINT"PHONE: ";:RUNT36
5
366 GOSUB97ØØ:GOSUB65Ø:IFCH$=""T
HEN365ELSE PH$=CH$
37Ø PRINT:PRINT"ENTER A PASSWORD
 OF UP TO 8":PRINT"CHARACTERS.
NOTE: UPPER AND":PRINT"LOWER CAS
E IS TREATED DIFFERENTLY."
371 PRINT:PRINT"PASSWORD: ";:RUN
T371
372 GOSUB97ØØ:GOSUB65Ø:IFCH$=""O
RLEN(CH$)>8THEN371ELSEPW$=CH$
375 PRINT:PRINT:PRINT
38Ø PRINT"I HAVE IT AS:"
385 PRINTTAB(5)NA$:PRINTTAB(5)AD
$:PRINTTAB(5)LO$:PRINTTAB(5)PH$
39Ø PRINT:PRINTTAB(5)"PASSWORD:
";PW$:PRINT:PRINT
395 PRINTTAB(5)"IS THIS CORRECT
 (Y/N)? ";
```

```
4ØØ GOSUB97ØØ:GOSUB6ØØ:IFCH$="Y"
ORCH$="y"THENPRINT"YES":GOTO43Ø
4Ø5 IFCH$="N"ORCH$="n"THENPRINT"
NO"ELSE4ØØ
41Ø PRINT"NAME: ";:RUNT41Ø
411 GOSUB97ØØ:GOSUB675:IFCH$=""T
HEN41ØELSENA$=CH$
415 PRINT"ADDRESS: ";:RUNT415
416 GOSUB65Ø:IFCH$=""THEN415ELSE
AD$=CH$
42Ø PRINT"CITY, STATE: ";:RUNT42
Ø
425 GOSUB97ØØ:GOSUB675:IFCH$=""T
HEN42ØELSELO$=CH$:GOTO36Ø
43Ø RUN T 46Ø:FR=FREE(Ø):IFFR<2T
HEN46Ø
45Ø OPEN"D",#2,"REGISTER/SYS",32
:FIELD#2,32 AS L$:K2=LOF(2):X$=S
TRING$(32,Ø):LSET L$=DA$+"  "+TI
$+X$:K2=K2+1:PUT#2,K2
455 K2=K2+1:LSET L$=NA$+X$:PUT#2
,K2:K2=K2+1:LSET L$=AD$+X$:PUT#2
,K2:K2=K2+1:LSET L$=LO$+X$:PUT#2
,K2:K2=K2+1:LSET L$=ZP$+X$:PUT#2
,K2:K2=K2+1:LSET L$=PH$+X$:PUT#2
,K2:K2=K2+1:LSET L$=PW$+X$:PUT#2
,K2
46Ø CLOSE:UNLOAD:GOSUB5ØØ
465 IFPEEK(4629)<3THEN985ØELSE21
Ø
5ØØ '-LSET NON REGISTERED USER-
5Ø5 GOSUB94ØØ:POKE4615,HR:POKE46
16,MN:POKE4617,Ø
51Ø GOSUB97ØØ:RUN T 9ØØ:GOSUB92Ø
Ø
515 LSET UN$=NA$+CHR$(Ø)+LO$+CHR
$(Ø):LSET UP$=PW$+STRING$(8,Ø)
52Ø LSET UR$=CHR$(Ø):LSET UA$=CH
R$(PEEK(466Ø)):LSET U1$=CHR$(PEE
K(4661)):LSET U2$=CHR$(PEEK(4662
))
525 LSET UU$=NU$:LSET UE$=CHR$(P
EEK(4664)):LSET U5$=MKN$(Ø):LSET
 UM$=MKN$(Ø)
53Ø LSET UD$=DT$:LSET U3$=NU$:LS
ET U4$=NU$:LSET UO$=CHR$(PEEK(46
63)):LSET UL$=CHR$(1)
535 LSET US$="    ":LSET SP$=""
54Ø IFFREE(PEEK(4572))<2THEN545E
LSE R=LOF(1)+1:PUT#1,R
545 RETURN
6ØØ '-SINGLE KEY ENTRY
6Ø5 EXEC4314:CH$=CHR$(PEEK(4481)
):IC$=INKEY$:IFIC$<>""THEN CH$=I
C$:GOTO615
61Ø GOSUB97ØØ:IF CH$=CHR$(Ø)THEN
6Ø5
615 RETURN
625 GOSUB6Ø5:CH=ASC(CH$)
```

```
63Ø IFCH>96ANDCH<123THEN CH=CH-3
2
635 CH$=CHR$(CH):RETURN
65Ø '-REMOTE INPUT  **KEYBOARD O
NLY
655 LINEINPUTCH$:GOSUB97ØØ
665 RETURN
675 GOSUB655:G$="":IFCH$=""THENR
ETURN
68Ø FOR A=1 TO LEN(CH$)
685 G=ASC(MID$(CH$,A,1)):IFG>96A
NDG<123THEN G=G-32
69Ø G$=G$+CHR$(G):NEXTA:CH$=G$:R
ETURN
7ØØ '-COMPARE FLAGS-
71Ø K7=Ø
72Ø FOR Q=1 TO 8
73Ø IFMID$(F$,Q,1)="Ø"THEN75Ø
74Ø IFMID$(PX$,Q,1)<>"1"THEN K7=
1
75Ø NEXTQ:RETURN
8ØØ '-SET TIME-
8Ø5 POKE65386,1Ø6:CLS:PRINT:PRIN
T:PRINT
81Ø INPUT"MM,DD,YY";MM,DD,YY
815 DC=VAL(HEX$(PEEK(&HØEFD))):P
OKE4611,DC
82Ø INPUT"HH:MM";HR,MN
825 X=USR1(HR*256+MN)
83Ø POKE46Ø8,MM:POKE46Ø9,DD:POKE
461Ø,YY
835 POKE65386,1Ø7
84Ø RETURN
9ØØ '-ERROR ROUTINE-
9Ø5 RUN T 9ØØ:CLOSE:ET=ET+1:IF E
T>9 THEN 99ØØ
91Ø '
915 ER=PEEK(&HFD):EL=PEEK(&HFE)*
&H1ØØ+PEEK(&HFF)
92Ø IF ER=>54 THEN EA=&HC242+ER:
GOTO935
925 IF ER=>5Ø THEN EA=&H88D9+ER:
GOTO935
93Ø EA=&HABAF+ER
935 ER$="**ERROR:  TYPE "+CHR$(P
EEK(EA))+CHR$(PEEK(EA+1))+" IN L
INE"+STR$(EL)+" :USER"
94Ø TR$=TR$+CHR$(13)+ER$:GOSUB96
ØØ
945 '
95Ø '
955 '
96Ø TR$=TR$+CHR$(13):GOSUB96ØØ
965 PRINT:POKE&HØ168,PEEK(4681):
POKE&HØ169,PEEK(4682)
97Ø PRINTER$:PRINT:FORA=1TO1ØØØ:
NEXTA
975 POKE&HØ168,&H1Ø:POKE&HØ169,&
HE6
98Ø RUN T 9ØØ:GOTO24
```

```
1ØØØ '-CD CKECK-
1ØØ5 IFPEEK(&HFF21)>1ØØTHEN RUN
1Ø1Ø RETURN
12ØØ REM
12Ø5 EXEC4314:IFPEEK(4481)<>42TH
ENRETURN
121Ø PRINT:PRINT"PLEASE PRESS EN
TER TO CONTINUE: ";
1215 GOSUB65Ø
1225 IF CH$=PW$ THEN II$=CHR$(95
):GOTO31
123Ø RETURN
41ØØ '-DECIMAL TO BINARY
41Ø5 F=ASC(F$):E=128:F$=""
411Ø FOR Q=1 TO 8
4115 J=INT(F/E)
412Ø IF J=Ø THEN F$=F$+"Ø"ELSEF$
=F$+"1"
4125 F=F-(E*J):E=E/2
413Ø NEXT Q
414Ø RETURN
42ØØ '-BINARY TO DECIMAL
42Ø5 W=LEN(F$):E=1:F=Ø
421Ø FOR Q=W TO 1 STEP -1
4215 IFMID$(F$,Q,1)="1"THEN F=F+
E
422Ø E=E*2:NEXTQ:F$=CHR$(F)
4225 RETURN
7ØØØ 'NEWUSER/POSTLOG/TXT
7ØØ5 IFPEEK(4667)<>ØTHEN259ELSEF
$="NEWUSER/TXT:":GOTO7Ø15
7Ø1Ø IFPEEK(4659)<>ØTHEN26ØELSEF
$="POSTLOG/TXT:"
7Ø15 D=-1:CLOSE:UNLOAD
7Ø2Ø D=D+1:IFD>3THEN7Ø45
7Ø25 FT$=F$+DR$(D):RUNT7Ø2Ø:OPEN
"I",#1,FT$
7Ø3Ø RUNT7Ø45:IFNOTEOF(1)THENLIN
EINPUT#1,S$ELSE7Ø45
7Ø35 EXEC&H1ØDA:CH$=CHR$(PEEK(44
81)):K=INSTR("pPSs",CH$):IFK>ØTH
EN7Ø5Ø
7Ø4Ø GOSUB97ØØ:PRINTS$:GOTO7Ø3Ø
7Ø45 CLOSE:UNLOAD:IFUT=1THEN26ØE
LSE259
7Ø5Ø IFK>2THEN7Ø45
7Ø55 EXEC&H1ØDA:IFPEEK(4481)<>13
THEN7Ø55ELSEPRINTS$
7Ø6Ø GOTO7Ø3Ø
8ØØØ '-SCREEN-
8ØØ5 RUN T 9ØØ
8Ø1Ø POKE&HØ168,PEEK(4681):POKE&
HØ169,PEEK(4682)
8Ø15 CLS3
8Ø2Ø GOSUB9ØØØ:IFK1=ØTHEN8Ø45
8Ø25 GET#1,1:RE=CVN(H1$)
8Ø3Ø GET#1,2:RS=CVN(H1$):XY=LOF(
1)-1
8Ø4Ø HR$=RIGHT$(STR$(PEEK(4615))
,2):MN$=RIGHT$(STR$(PEEK(4616)),
2):TZ$=HR$+":"+MN$
8Ø41 FOR Q=1 TO 4
8Ø42 IFMID$(TZ$,Q,1)=" " THEN MI
D$(TZ$,Q,1)="Ø"
8Ø43 NEXTQ
8Ø45 CLOSE
8Ø5Ø GOSUB92ØØ:GET#1,1
8Ø55 K=INSTR(UN$,CHR$(Ø)):N$=LEF
T$(UN$,K-1):L=INSTR(K+1,UN$,CHR$
(Ø)):L$=MID$(UN$,K+1,L-1-K):X=AS
C(U7$)*256+ASC(U8$):CA=LOF(1)-1
8Ø6Ø PRINT@66,"COLOR BULLETIN BO
ARD SYSTEMS";
8Ø65 PRINT@14Ø,"<C> 1984";:PRINT
@167,"by RICHARD  DUNCAN";
8Ø7Ø PRINT@228,"STARTING MESSAGE
:";RS;:PRINT@26Ø,"  ENDING MESSA
GE:";RE;:PRINT@328,"CALLS: ";X;:
PRINT@36Ø,"USERS: ";CA;
8Ø75 NN$="LAST CALLER: "+N$:A=LE
N(NN$):IFA>32THEN B=Ø:GOTO8Ø8Ø E
LSE B=INT((32-A)/2)
8Ø8Ø GOSUB94ØØ:PRINT@448+B,NN$;:
PRINT@49Ø,"TIME: ";TZ$;
8Ø85 POKE&HØ168,&H1Ø:POKE&HØ169,
&HE6
8Ø9Ø CLOSE:RETURN
81ØØ '-SYSTEM CONTROL FILE-
81Ø5 DSKI$ Ø,17,18,S1$,S2$:IFLEF
T$(S1$,5)<>"COBBS"THEN9925
811Ø POKE4629,ASC(MID$(S1$,9,1))
8111 POKE4664,ASC(MID$(S1$,1Ø,1)
)
8112 POKE4663,ASC(MID$(S1$,11,1)
)
8113 FOR A=Ø TO 2:POKE466Ø+A,ASC
(MID$(S1$,12+A,1)):NEXT A
8114 POKE4628,ASC(MID$(S1$,15,1)
)
8115 POKE4666,ASC(MID$(S1$,16,1)
)
8116 POKE4669,ASC(MID$(S1$,17,1)
)
8117 POKE4679,ASC(MID$(S1$,18,1)
)
8118 POKE468Ø,ASC(MID$(S1$,19,1)
)
8119 POKE4667,ASC(MID$(S1$,2Ø,1)
)
812Ø POKE4659,ASC(MID$(S1$,21,1)
)
8121 POKE4694,ASC(MID$(S1$,22,1)
)
8122 FORA=ØTO6:POKE467Ø+A,ASC(MI
D$(S1$,23+A,1)):NEXT A
8199 RETURN
82ØØ '-MENU NAMES-
82Ø5 GOSUB91ØØ
821Ø FOR R=1 TO K2 STEP 4
```

```
8215 GET#2,R:BN$=M5$:BN=ASC(M1$)
822Ø K=INSTR(BN$,NU$):IFK=ØTHEN
K=LEN(BN$)+1:BN$(BN)=LEFT$(BN$,K
-1)
8225 BN$(BN)=LEFT$(BN$,K-1):NEXT
 R:BN$(255)="SYSOP MSG"
823Ø CLOSE:RETURN
9ØØØ '-OPEN HDR/SYS-
9ØØ5 RUNT99ØØ:FF$="HDR/SYS:"+DR$
(PEEK(467Ø))
9Ø1Ø OPEN"D",#1,FF$,11Ø
9Ø15 FIELD#1,5 AS H1$,1 AS H2$,3
 AS H3$,2 AS H4$,8Ø AS H5$,5 AS
H6$,5 AS H7$,1 AS H8$,8 AS SP$
9Ø2Ø K1=LOF(1):RETURN
91ØØ '-OPEN BOARD MENU-
91Ø5 FF$="MENU/SYS:"+DR$(PEEK(46
73))
911Ø OPEN"D",#2,FF$,25Ø
9115 FIELD#2,1 AS M1$,1 AS M2$,1
 AS M3$,1 AS M4$,16 AS M5$,23Ø A
S M6$
912Ø K2=LOF(2):RETURN
92ØØ '-OPEN USERLOG-
92Ø5 FF$="USERL/SYS:"+DR$(PEEK(4
672))
921Ø OPEN"D",#1,FF$,96
9215 FIELD#1,5Ø AS UN$,8 AS UP$,
1 AS UR$,1 AS UA$,1 AS U1$,1 AS
U2$, 1 AS UU$,1 AS UE$,5 AS U5$,
5 AS UM$,5 AS UD$,1 AS U3$,1 AS
U4$,1 AS UO$,1 AS UL$,4 AS US$,1
 AS U7$,1 AS U8$,7 AS SP$
922Ø KX=LOF(1):K1=KX-1:RETURN
94ØØ '-GET TIME-
94Ø5 DC=VAL(HEX$(PEEK(&HØEFD))):
HR=VAL(HEX$(PEEK(&HØEFE)))
941Ø MN=VAL(HEX$(PEEK(&HØEFF))):
SS=VAL(HEX$(PEEK(&HØFØØ)))
9415 HR$=RIGHT$(STR$(HR),2):IFHR
<1ØTHEN MID$(HR$,1,1)="Ø"
942Ø MN$=RIGHT$(STR$(MN),2):IFMN
<1ØTHEN MID$(MN$,1,1)="Ø"
9425 TI$=HR$+":"+MN$:IF SS<57 OR
 MN>58  THEN 944Ø
943Ø MN=MN+1:IF MN>59 THEN MN=Ø:
HR=HR+1:IF HR>23 THEN HR=Ø
9435 X=USRØ(HR*256+MN)
944Ø X=PEEK(4611):IF X<>DC THEN
9455
9445 DA$=RIGHT$(STR$(PEEK(46Ø8))
,2)+"/"+RIGHT$(STR$(PEEK(46Ø9)),
2)+"/"+RIGHT$(STR$(PEEK(461Ø)),2
)
945Ø RETURN
9455 MM=PEEK(46Ø8):DA=PEEK(46Ø9)
:YY=PEEK(461Ø):DA=DA+1:IFDA>31TH
EN DA=1:MM=MM+1:IF MM>12 THEN MM
=1:YY=YY+1
```

```
946Ø POKE46Ø8,MM:POKE46Ø9,DA:POK
E461Ø,YY:POKE4611,DC:GOTO9445
96ØØ '-TRACER-
96Ø5 CLOSE:FF$="TRACER/SYS:"+DR$
(PEEK(4675))
961Ø FR=PEEK(4669):IFFR=ØTHEN962
5ELSEIFFR=1THEN964ØELSEIFFR=3THE
N966ØELSERUNT963Ø:OPEN"D",#1,FF$
,128
9615 FIELD#1,128 AS X$:K1=LOF(1)
962Ø LSET X$=TR$:K1=K1+1:PUT#1,K
1
9625 CLOSE:TR$="":RETURN
963Ø POKE4669,Ø:GOTO295
964Ø OPEN"O",#-1,"TRACER":WRITE#
-1,TR$:GOTO9625
966Ø PT=PEEK(&HFF22)AND1:IFPT=1T
HEN9625ELSEPRINT#-2,TR$:GOTO9625
97ØØ '-CD CHECK-
97Ø5 IFPEEK(4658)=ØTHEN974Ø
971Ø CD=PEEK(65385)AND32
9715 IF CD<>Ø OR PEEK(4657)<>Ø T
HEN975Ø
974Ø RETURN
975Ø CLOSE:RUN
98ØØ '-TERMINATE-
98Ø5 GOSUB96ØØ
981Ø CLOSE:POKE65386,1Ø6:RUN
985Ø PRINT"THANKS FOR CALLING Co
BBS.":PRINT"WE WILL CONTACT YOU
SHORTLY"
9855 PRINT"TO UPGRADE YOUR ACCES
S."
986Ø GOTO98ØØ
99ØØ '-UNABLE TO ACCESS-
99Ø4 GOTO9ØØ
99Ø5 H$="PASSWORD"
991Ø POKE65386,1Ø6:FORT=1TO1ØØ:N
EXTT
9915 POKE65386,1Ø7
992Ø IFPEEK(65385)AND32<>ØTHEN99
2Ø
9925 FOR A=1 TO 5:PRINT:NEXTA
9926 GOTO9ØØ
993Ø PRINT"THANKS FOR CALLING...
":PRINT:PRINT"WE ARE UNABLE TO A
CCESS THE"
9935 PRINT"SYSTEM PROPERLY AT TH
IS TIME.":FORT=1TO5ØØ:NEXTT
994Ø PRINT:PRINT:PRINT
9945 PRINT"PLEASE CALL AGAIN LAT
ER.":PRINT:FORT=1TO1ØØØ:NEXTT
995Ø PRINT"PLEASE HANG UP.":PRIN
T:PRINT:GOSUB675:IFCH$=H$ THEN I
I$=CHR$(95):GOTO31
9955 GOTO99ØØ
1ØØØØ POKE65386,1Ø6
1ØØØ5 IFPEEK(4684)<>188THENPOKE&
HAC7D,&HØE:POKE&HAC7E,&HØØ
1ØØ15 MOTOROFF
1ØØ2Ø X=&H112B:POKE X,&H86:POKE
X+1,&HØD:POKE X+2,&HB7:POKE X+3,
&H11:POKE X+4,&H81:POKE &H1127,&
H1Ø
1ØØ25 RUN
```

```
100 ......148
210 ......150
300 ......203
480 ......246
END ......87
```

Listing 3: COTERM

```
1Ø '-CREATES COTERM/BIN FOR COBB
S
2Ø DATA 2Ø4,126,57,231,14Ø,25Ø,4
8,141,Ø,21,246,1,148,254,1,149,2
31,141,Ø,3Ø
3Ø DATA 239,141,Ø,27,183,1,148,1
91,1,149,57,31,169,53,64,129,84,
39,16,52
4Ø DATA 64,52,4,198,57,247,1,142
,53,1,126,255,255,255,255,157,15
9,189,175,1Ø3
5Ø DATA 198,57,247,1,142,22Ø,43,
39,21,189,173,1,37,17,175,141,25
5,231,48,141
6Ø DATA Ø,12,191,1,143,198,126,2
47,1,142,57,126,174,21Ø,134,57,1
83,1,142,158
7Ø DATA 1Ø4,48,1,39,241,215,253,
22Ø,1Ø4,221,254,189,173,51,174,1
41,255,191,189,174
8Ø DATA 187,126,173,158,255,Ø,25
5,Ø,Ø,77,39,5,2Ø4,255,255,32,65,
189,179,237
9Ø DATA 77,38,245,193,3,34,241,2
47,14,Ø,142,Ø,234,236,129,52,6,1
4Ø,Ø,24Ø
1ØØ DATA 38,247,182,14,Ø,151,235
,142,16Ø,Ø,159,238,134,2,151,234
,48,141,Ø,7
11Ø DATA 52,16,5Ø,122,189,214,11
2,142,Ø,234,53,6,237,129,14Ø,Ø,2
4Ø,38,247,23Ø
12Ø DATA 132,79,126,18Ø,244,Ø,25
5,Ø,22,Ø,145,22,Ø,159,22,1,9,22,
1,178
13Ø DATA 79,1Ø6,141,Ø,32,46,1ØØ,
48,141,Ø,26,134,12Ø,167,132,166,
13Ø,45,88,139
14Ø DATA 1,25,167,132,161,6,45,2
8,111,132,32,239,255,255,255,255
,255,Ø,Ø,7
15Ø DATA 36,96,96,Ø,255,Ø,255,Ø,
1Ø6,14Ø,242,141,2Ø3,11Ø,156,243,
23Ø,14Ø,23Ø,88
16Ø DATA 88,48,141,Ø,43,58,198,4
```

```
,2Ø6,4,2Ø,189,165,154,48,14Ø,213
,141,1Ø,134
17Ø DATA 58,167,192,141,4,134,58
,167,192,166,132,68,68,68,68,138
,48,167,192,166
18Ø DATA 128,132,15,138,48,167,1
92,57,19,21,14,32,13,15,14,32,2Ø
,21,5,32
19Ø DATA 23,5,4,32,2Ø,8,18,32,6,
18,9,32,19,1,2Ø,32,189,179,237,7
7
2ØØ DATA 45,3,231,14Ø,144,79,23Ø
,14Ø,14Ø,189,18Ø,244,57,14Ø,32,5
5,19Ø,255,248,238
21Ø DATA 1,239,14Ø,135,51,14Ø,13
7,239,1,52,2,182,255,3,138,1,183
,255,3,2Ø6
22Ø DATA 1,1Ø6,166,196,174,65,16
7,141,255,111,175,141,255,1Ø8,13
4,126,48,141,Ø,148
23Ø DATA 167,196,175,65,134,18,1
67,14Ø,2ØØ,53,2,189,179,237,77,4
5,22,52,4,141
24Ø DATA 26,231,141,255,65,53,2,
141,18,231,141,255,58,111,141,25
5,55,28,239,236
25Ø DATA 141,255,47,189,18Ø,244,
57,95,14Ø,2Ø3,16,128,1Ø,44,25Ø,1
39,1Ø,52,2,234
26Ø DATA 224,57,189,179,237,52,4
,141,234,231,141,Ø,6Ø,53,2,141,2
26,231,141,Ø
27Ø DATA 53,26,16,182,255,32,133
,1,38,249,142,7,2Ø8,48,31,39,44,
182,255,32
28Ø DATA 133,1,39,245,23,254,2Ø5
,236,141,254,235,16,163,141,Ø,17
,39,17,173,159
29Ø DATA 16Ø,Ø,39,215,31,137,79,
189,18Ø,244,28,239,57,Ø,255,79,9
5,189,18Ø,244
3ØØ DATA 57,2Ø4,255,255,189,18Ø,
244,57,5Ø,98,23,254,2Ø4,15,112,1
3,111,16,38,Ø
31Ø DATA 7Ø,52,2Ø,174,141,254,18
8,191,1,13,134,239,167,159,Ø,136
,182,255,32,133
32Ø DATA 1,38,249,142,7,2Ø8,48,3
1,39,31,182,255,32,133,1,39,245,
23,254,112
33Ø DATA 173,159,16Ø,Ø,39,226,19
8,96,231,159,Ø,136,48,141,254,14
4,191,1,13,53
34Ø DATA 148,48,141,254,135,191,
1,13,126,161,179,126,161,127,Ø,2
55,67,48,141,Ø
35Ø DATA 81,188,1,1Ø4,39,5Ø,182,
1,1Ø3,167,141,Ø,218,19Ø,1,1Ø4,17
5,141,Ø,212
36Ø DATA 182,1,1Ø6,167,141,Ø,2Ø7
,19Ø,1,1Ø7,175,141,Ø,2Ø1,134,126
,183,1,1Ø6,183
37Ø DATA 1,1Ø3,48,141,Ø,36,191,1
,1Ø4,48,141,Ø,4,191,1,1Ø7,57,15,
112,13
38Ø DATA 111,16,38,Ø,169,127,255
,64,5Ø,98,141,44,129,3,38,2,134,
42,183,17
39Ø DATA 129,57,52,2,18,18,18,18
,18,18,18,18,18,18,15Ø,111,53,2,
16,38
4ØØ DATA Ø,129,141,62,18,18,18,1
8,18,18,18,18,18,18,32,115,52,21
,26,8Ø
41Ø DATA 173,159,16Ø,Ø,39,2,32,3
6,182,255,1Ø5,132,8,39,6,182,255
,1Ø4,183,17
42Ø DATA 129,246,255,1Ø5,196,32,
39,3,247,18,49,53,149,18,18,18,1
8,18,18,18
43Ø DATA 18,18,18,18,53,149,52,2
3,26,8Ø,246,255,1Ø5,196,32,38,35
,246,255,1Ø5
44Ø DATA 196,16,39,242,183,255,1
Ø4,129,13,38,21,246,255,1Ø5,196,
32,38,14,246,255
45Ø DATA 1Ø5,196,16,39,242,134,1
Ø,183,255,1Ø4,18,18,18,18,18,18,
246,255,1Ø5,196
46Ø DATA 32,39,3,247,18,49,53,15
1,18,18,18,126,2Ø3,74,126,197,14
3,13,8Ø,1
47Ø DATA 1Ø3,134,68,69,86,78,85,
77,Ø,111,Ø,255,Ø,255,Ø,255,Ø,255
,Ø,255
48Ø DATA Ø,255,Ø,255,Ø,255,Ø,255
,Ø,255,Ø,255,Ø,255,Ø,255,Ø,255,Ø
,255
49Ø DATA Ø,255,Ø,255,Ø,255,Ø,255
,Ø,255,Ø,255,Ø,255,Ø,255,Ø,255,Ø
,255
5ØØ DATA Ø,255,Ø,255,Ø,255,Ø,255
,Ø,255,Ø,255,Ø,255,Ø,255,Ø,255,Ø
,255
51Ø DATA Ø,255,Ø,255,Ø,255,Ø,255
,Ø,255,Ø,255,Ø,255,Ø,255,Ø,255,Ø
,255
52Ø DATA Ø,255,Ø,255,Ø,255,Ø,255
,Ø,255,Ø,255,Ø,255,Ø,255,Ø,255,Ø
,255
53Ø DATA Ø,255,Ø,255,248
54Ø DATA END
55Ø A= 3584
56Ø READ B$:IF B$="END" THEN 58Ø
57Ø POKE A,VAL(B$):A=A+1:GOTO  5
6Ø
58Ø PRINT"SAVING  COTERM/BIN"
59Ø SAVEM"COTERM/BIN",&HØEØØ,&H1
22Ø,&H1Ø91
```

AT THE CENTRE OF COCO'S WORLD

by Jack Fricker

(Jack is just back from a trip to Rainbowfest in Princeton, and to the Microware Users' Conference in Iowa. G.)

A new release of CoCo-OS9 is about to hit the streets. It should be hitting the US about the time you read this and be seen here soon after, probably in the new year.

Fortunately I was able to get a look at the new release (Version 2.0.0) at the Microware seminar at Des Moines, Iowa, - in November. Unfortunately some of the same old problems relating to the limiting of disk drive space (35 track single sided only) remain. These drivers have been radically changed and the present replacements will not work. However, the good news is there are some software vendors working on the problem so by the time it is released there should already be replacement modules available.

One of the BIG news items to come out of the seminar is the OFFICIAL support for bard disks by TANDY. The drives supported are the 15 and 35 megabyte Winchesters sold by TANDY to work on the Model 4. You are limited to 2 of these drives and they MUST be the same size.

There are also other suppliers who make 10 & 20 meg hard disk drives available now. I bought a 10 meg at Princeton Rainbowfest. These cost between $US500 and $US900. You can double that price by the time you get it here!

The good news is that if you do buy one of these "third party" drives and modules, some of these allow you to mix and match up to eight (8) hard disk drives to your CoCo. At present sizes this could mean up to 160 MEGABYTES of permanent storage. Of course you would never use anywhere near that number of drives, but adding a hard drive speeds up the operation of your CoCo considerably.

The new version of OS9 will not boot directly from the hard disk so you must still have one of your floppy drives connected. When 2.0 first boots it looks for the hard disk drivers and if found changes automatically to them, otherwise if not found will go back to the floppy for all commands.

There have been a number of other major changes, one of these is that OS-9 now checks for an 80 column display. This has just been done on Tandy's instruction because they MAY release an 80 col card. However they have no plans at present and the 80 col card from them may never see the light of day.

Again the third party 80 col cards (about $Aust200) will work. Some of the utilities will now check the width of the screen as they are run and adjust themselves to suit. Two screen sizes are supported. These are the standard 32 col display and the new 80 col display.

One other thing is that the 80 col card gives you more user memory than the 32 col display does. It doesn't use the CoCo's memory because it contains it's own memory. This is very like one of the more popular third party cards.

For those of you who have the new version of the CoCo (marked TANDY instead of Radio Shack), version 2.0 will come up with the lower case enabled. If you try to enable the lower case and it is not fitted, the VDG will display one of the semi-graphics screens that you can't read and it will look as though it has crashed.

Auto Key Repeat has also been added as well and a Control G will cause the terminal to beep at you. The four additional function keys that some of us have fitted to the empty address space in the keyboard matrix have been decoded and will now give ASCII values when pressed.

Again this is because Tandy DID produce a keyboard that had the function keys fitted but never released it. They sold them off cheaply and didn't tell anyone that they fitted the CoCo. This keyboard also had the arrow keys set in a diamond shaped cluster. I saw one in a shop here for about $30 or so but didn't think of putting it in a CoCo.

There are also new graphic commands that have been added. I was not impressed and thought much more could have been done to simplify the making and storing of graphic screens which is at present more difficult than in Basic.

The RS232 driver bug has also been fixed. What do you mean you didn't know there was a bug? Well to tell the truth neither did I.

The problem is that you cannot use the T1 terminal without using the T2 module and the Tandy RS232 cartridge as well. When mine didn't work properly I thought that I was doing something wrong and went back to using version 1.0.

With the version 2.0 it makes no difference whether the ACIAPAK is connected or not.

Speaking of the communications PAK's, Tandy are also about to release a new ACIA cartridge called MODPAK, which in fact contains 2 of them. These are limited to 300 baud each when used with their existing ACIA cartridge and so are ideal for use with modems.

Or they can be modified to replace the ACIAPaK and then one can run at 19200 baud and the other at 300.

Rumours were flying thick and fast about a new TANDY 6809 machine which will probably be upwardly compatable with the CoCo. This time it is supposed to be an OS-9 level 2 machine which will mean it runs faster and will have more memory and RAM disks. Of course this also may never see the light of day and at present Tandy at Fort Worth will neither confirn or deny it.

For those of you who don't know what a RAM disk is , it is a section of memory that can be set up to act as though it were a disk drive. The advantage of this is that when you first boot your system you can copy your commands disk to your RAM disk and when you want a command or file it is already in memory and so is executed much faster than even a hard drive.

Speaking of RAM disks, Spectrum Project's Thunder-RAM memory board went over very well at the Rainbowfest in Princeton. There is a strong probability that we will see a memory board available as a plug-in cartridge, which will make it more portable to different versions of the CoCo.

RAINBOWTECH

OS-9

KISSable OS-9

Confessions Of An Enlightened Spreadsheet User

By Dale L. Puckett

Getting Started with *DynaCalc*

It's time now for True Confessions. I've worked with microcomputers since the Southwest Technical Products 6800 machine was introduced. I've learned a little BASIC and taught myself assembly language programming, but in the past when a person showed me a spreadsheet I would panic. I don't know what caused my mental block, but fortunately everything changed when I had to sit down and convert a spreadsheet at work. Our branch manages the Pollution Response Program for the entire Coast Guard and this means we manage the resources of the national strike force. The three strike teams send in data each quarter that fits nicely in a spreadsheet.

Recently, we decided to start collecting the same type of data about the Public Information Assist Team I supervise. The idea was great but the information was different. This meant I had to learn to edit a *Multiplan* spreadsheet. After a quick glance at the book and a few experiments at a Coast Guard C-3 standard terminal, I actually made it work.

Now for the good news. The experience with *Multiplan* made me brave and I came home one night and dived into *DynaCalc*. Guess what? It's just as easy to run, just as fast and just as powerful. And most importantly, since we have been preaching that "OS-9 needs excellent application programs" for months, yet only writing about system programs and languages, I decided it was time to delve into this aspect. So, this month "Dale does *DynaCalc*."

Comments from Joe Turner

DynaCalc is very easy to use. In fact, it's so easy to use that Joe Turner at Computer Systems Center in Chesterfield, Mo., rarely gets any calls. Unfortunately, that changed shortly after Tandy released the OS-9 version of *DynaCalc* for the Color Computer. Yet Joe has only had to answer one question: "Why doesn't *DynaCalc* work right with the *Word-Pak* 80-column card from PBJ?" The solution: Contact PBJ and have them send you Version 3.1 or higher of their OS-9 *Word-Pak* drivers.

The problem is in the getstat call, which reads the screen size. The original Color Computer OS-9 didn't have one, but when Frank Hogg designed the drivers for his Hi-Res screen, he added it. Later, he worked closely with PBJ's software wizard Ed Bender to standardize the call. The result is Hogg's *Dyna-*

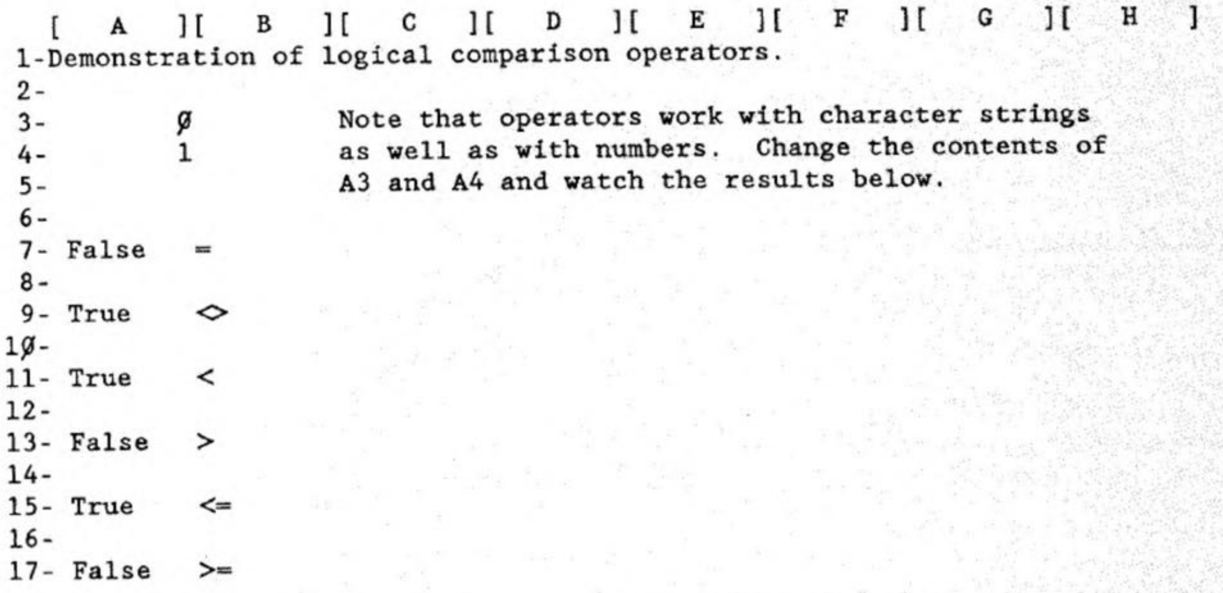
Star text editor works automatically and equally well using a Hi-Res screen or a PBJ 80-column card.

DynaCalc needs an erase to end of line code, a cursor on and off code, and a screen size call. That's one of the reasons OS-9 Version 1.01 was released. But unfortunately, when Tandy implemented the getstat call and the clear screen call they did it differently, i.e., the code is in the 6809's B Register instead of the A Register or vice versa. Also, an offset may be different. Since the Color Computer version of *DynaCalc* was written for the Tandy drivers, it does not work properly with earlier versions of the *Word-Pak* software. Give Al at PBJ a call for the latest version of the *Word-Pak* drivers.

About the only other question Turner gets concerns the missing /o command that is available in other versions of *DynaCalc*, but not the Color Computer. And, there's a good reason — the command isn't needed because of OS-9's unified I/O. On the Color Computer, if you want to save a listing of the spreadsheet, you just print it to a file. You do this by typing /spc, which closes the OS-9 output path to your printer device descriptor /p. Then, the next time you type /p to print a spreadsheet, *DynaCalc* will ask for a filename. After you open this path, *DynaCalc* will send all printouts to a file until you type /spc again to close the file. That's the way the four listings featured in this column were produced.

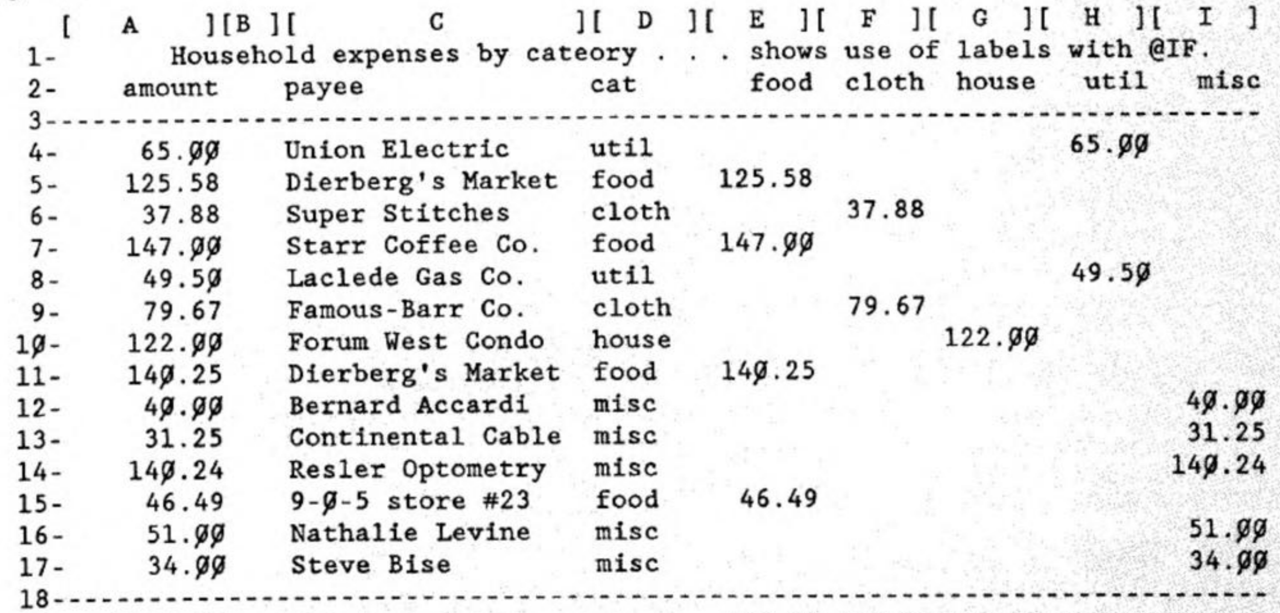
The present version of OS-9 *DynaCalc* does not have graphics. Will it in the future? Probably not. Will you be able to produce graphics from the *DynaCalc* spreadsheets in the future? Probably yes. How? By executing the code in a new *DynaCalc* filter module.

Figure 1



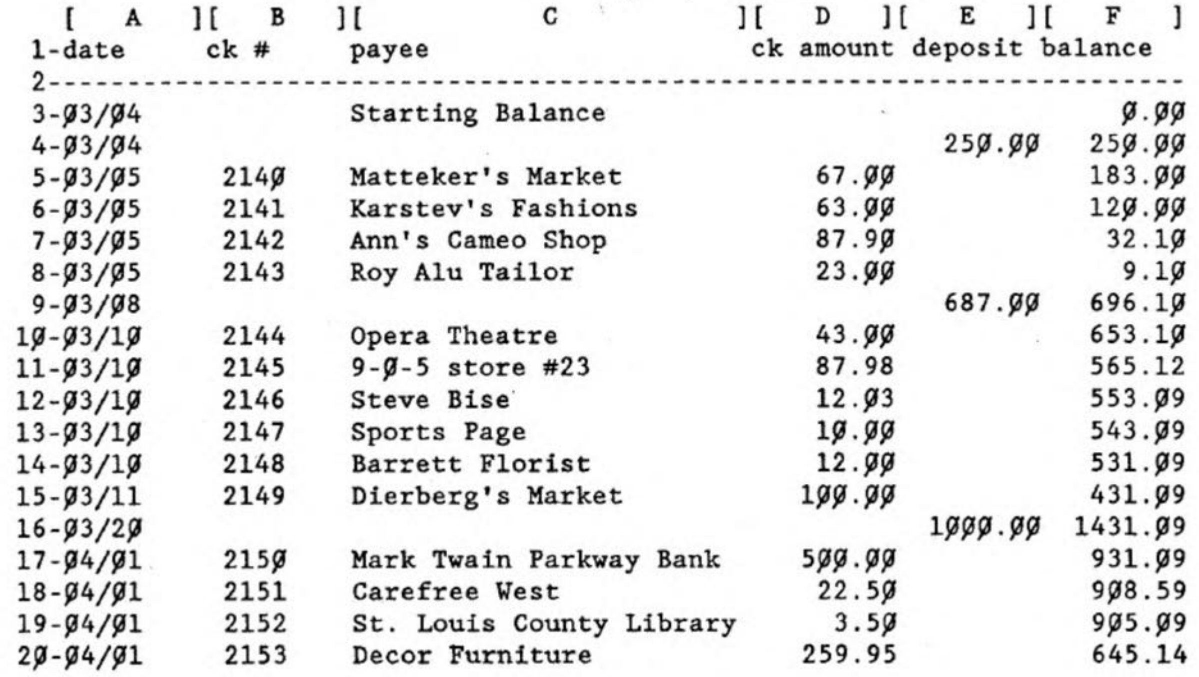
```
   [  A   ][  B  ][  C  ][  D  ][  E  ][  F  ][  G  ][  H  ]
 1-Demonstration of logical comparison operators.
 2-
 3-        Ø           Note that operators work with character strings
 4-        1           as well as with numbers.  Change the contents of
 5-                    A3 and A4 and watch the results below.
 6-
 7- False   =
 8-
 9- True    <>
1Ø-
11- True    <
12-
13- False   >
14-
15- True    <=
16-
17- False   >=
```

Figure 2



```
  [  A    ][B ][       C         ][ D  ][ E  ][ F  ][ G  ][ H  ][ I  ]
1-       Household expenses by cateory . . . shows use of labels with @IF.
2-   amount    payee               cat      food  cloth  house   util   misc
3--------------------------------------------------------------------------
4-     65.ØØ   Union Electric      util                          65.ØØ
5-    125.58   Dierberg's Market   food   125.58
6-     37.88   Super Stitches      cloth         37.88
7-    147.ØØ   Starr Coffee Co.    food   147.ØØ
8-     49.5Ø   Laclede Gas Co.     util                          49.5Ø
9-     79.67   Famous-Barr Co.     cloth         79.67
1Ø-   122.ØØ   Forum West Condo    house                122.ØØ
11-   14Ø.25   Dierberg's Market   food   14Ø.25
12-    4Ø.ØØ   Bernard Accardi     misc                                 4Ø.ØØ
13-    31.25   Continental Cable   misc                                 31.25
14-   14Ø.24   Resler Optometry    misc                                14Ø.24
15-    46.49   9-Ø-5 store #23     food    46.49
16-    51.ØØ   Nathalie Levine     misc                                 51.ØØ
17-    34.ØØ   Steve Bise          misc                                 34.ØØ
18-------------------------------------------------------------------------
19-  11Ø9.86   totals                     459.32 117.55 122.ØØ 114.5Ø 296.49
2Ø-                                                                        ok
21-          ok bad
```

Figure 3



```
    [  A   ][  B   ][            C             ][  D   ][  E   ][  F   ]
  1-date      ck #     payee                     ck amount deposit balance
  2----------------------------------------------------------------------
  3-Ø3/Ø4              Starting Balance                              Ø.ØØ
  4-Ø3/Ø4                                                250.ØØ    250.ØØ
  5-Ø3/Ø5     214Ø     Matteker's Market            67.ØØ          183.ØØ
  6-Ø3/Ø5     2141     Karstev's Fashions           63.ØØ          12Ø.ØØ
  7-Ø3/Ø5     2142     Ann's Cameo Shop             87.9Ø           32.1Ø
  8-Ø3/Ø5     2143     Roy Alu Tailor               23.ØØ            9.1Ø
  9-Ø3/Ø8                                                687.ØØ    696.1Ø
 1Ø-Ø3/1Ø     2144     Opera Theatre                43.ØØ          653.1Ø
 11-Ø3/1Ø     2145     9-Ø-5 store #23              87.98          565.12
 12-Ø3/1Ø     2146     Steve Bise                   12.Ø3          553.Ø9
 13-Ø3/1Ø     2147     Sports Page                  1Ø.ØØ          543.Ø9
 14-Ø3/1Ø     2148     Barrett Florist              12.ØØ          531.Ø9
 15-Ø3/11     2149     Dierberg's Market           1ØØ.ØØ          431.Ø9
 16-Ø3/2Ø                                               1ØØØ.ØØ   1431.Ø9
 17-Ø4/Ø1     215Ø     Mark Twain Parkway Bank     5ØØ.ØØ          931.Ø9
 18-Ø4/Ø1     2151     Carefree West                22.5Ø          9Ø8.59
 19-Ø4/Ø1     2152     St. Louis County Library      3.5Ø          9Ø5.Ø9
 2Ø-Ø4/Ø1     2153     Decor Furniture             259.95          645.14
```

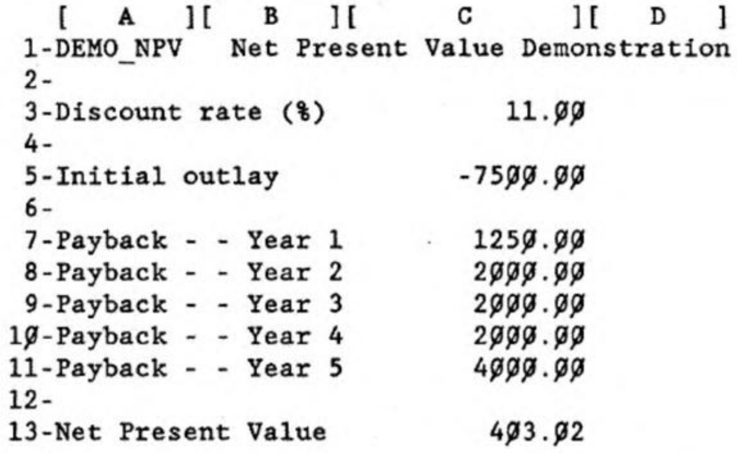
You will most likely see a filter that reads its input from a standard OS-9 data file output in text form by *DynaCalc*. Its output will most likely produce charts and graphs on a standard Color Computer OS-9 graphics screen. Of course, when OS-9 Level II hits the CoCo there will be enough memory available to let you run this filter from within *DynaCalc* using the built-in system execution command. Again, that's one of the features that makes the OS-9 operating system shine.

Before we look at several *DynaCalc* examples let's see if we can get you off to a good start psychologically by borrowing a philosophy used on Apple's Macintosh. Essentially, every command in a Macintosh menu is a verb. It doesn't matter what you are doing, you run every program the same way. First, you select something — a character, a word, a sentence, a spreadsheet cell or an entire row or column, then what you do is determined by the action verb in the menu.

I found that thinking about *DynaCalc* in this manner made it very easy to understand. In a nutshell, you point to a cell by moving the cursor around with the arrow keys. Then, do something to it. Sometimes you type in original data, sometimes you type in an expression or formula and other times you just change the value of the variable stored in the cell selected.

Since *DynaCalc* is versatile, it lets the user enter data in many ways. For example, you can move to a cell by

Figure 4



```
    [  A   ][  B   ][    C    ][  D   ]
  1-DEMO_NPV   Net Present Value Demonstration
  2-
  3-Discount rate (%)             11.ØØ
  4-
  5-Initial outlay            -75ØØ.ØØ
  6-
  7-Payback - - Year 1          125Ø.ØØ
  8-Payback - - Year 2          2ØØØ.ØØ
  9-Payback - - Year 3          2ØØØ.ØØ
 1Ø-Payback - - Year 4          2ØØØ.ØØ
 11-Payback - - Year 5          4ØØØ.ØØ
 12-
 13-Net Present Value            4Ø3.Ø2
```

Listing 1a:

```

t
tmode .1 -pause
load dir
rename /dØ/cmds/dir Dir32
debug dir <DirScript
save temp dir
unlink dir
verify u <temp >/dØ/cmds/Dir
attr /dØ/cmds/dir e pe
del temp
tmode .1 pause
-t

********************************
```

Listing 1b:

```
1 dir
. .+3
=A9
. .+4
=83
1 dir
=2Ø
=2Ø
=ØD
=ØA
1 dir
. .+137
=32
```

```
 . .+37
=4E
=41
=4D
=45
=2Ø
=2Ø
 . .+2
=2Ø
=2Ø
=2Ø
=2Ø
 . .+5
=2Ø
=2Ø
=2Ø
=2Ø
=2Ø
 . .+6
=2Ø
=2Ø
=2Ø
=2Ø
=2Ø
=2Ø
=41
=54
=54
=52
=2Ø
=2Ø
=2Ø
=53
=54
=41
=52
=54
=2Ø
=2Ø
=2Ø
=2Ø
=2Ø
=2Ø
=53
=49
=5A
=45
=ØA
=ØD
=2Ø
=2Ø
=2Ø
```

```
. .+3
=1Ø
. .+35
=Ø1
=22
1 dir
. .+295
=34
=Ø4
=D6
=Ø8
=CØ
=5C
=C1
=32
=27
=Ø5
=17
=FF
=3D
=2Ø
=F3
=35
=84
q

**********************************
```

Listing 2a:

```
t
tmode .1 -pause
load mdir
rename /dØ/cmds/mdir Mdir32
debug mdir <MDirScript
save temp mdir
unlink mdir
verify u <temp >/dØ/cmds/Mdir
attr /dØ/cmds/mdir e pe
del temp
tmode .1 pause
-t

**********************************
```

Listing 2b:

```
1 mdir
. .+C1
=32
. .+3
=1Ø
q

**********************************
```

Listing 3:

```
PROCEDURE pixscroll
 ØØØØ        (* procedure to search a dir for graphic pictures *)
 ØØ34        (* and then to scroll thru them on the screen *)
 ØØ64        (* by Mark W. Smith *)
 ØØ7A
 ØØ7B        (* all pix should be copied to OS9, e.g. xcopy *)
 ØØAC        (* save them with an ext of '.pix' *)
```

pressing the arrow keys until the cursor arrives at your destination. Or, go directly to a cell by using *DynaCalc's* GoTo command, the '>' character. Or, if you don't remember the row column address of the cell, but do remember what it contains, you can use a special command to look for the contents and move the cursor to it automatically.

When *DynaCalc* needs parameters, you can communicate them in several different ways. For example, when you want to find the total value of a column of numbers, you may arrive at the answer following several paths. Pick the approach that makes sense to you. The first time I tried to run a spreadsheet, I went to the cell where I wanted to hold the total and typed `+A1+A2+A3`, etc. Believe me, that's the hard way but it works if you don't mind the grunt work.

Enter a standard *DynaCalc* function @SUM. Now, I go to the cell where I want to store the total and type `@SUM(A1 . . . A14`. And, you only need to type the first period; *DynaCalc* adds the other two periods in the ellipse.

However, there's an easier way. To type the expression mentioned, I had to know that my data started in the cell named "A1" and ended in the cell named "A14." It's easier to point to the cell where you want to store the result and type `@SUM(`. Then, move the cursor until it points to the first cell you want to add in the total and hit RETURN. Now, move the cursor until it points to the last cell in the range and hit RETURN. If you still can't believe the magic, look at the left end of the second line on your *DynaCalc* screen. Notice the expression you just "typed?" Amazing!

Turner gave us permission to print several sample spreadsheets that are supplied on the disk with larger OS-9 systems. After studying them for a few minutes, you'll be amazed at what can be done with a spreadsheet. In Turner's words, "It bears repeating. People that are programmers can knock out a program fast. But, they can often solve the same problem much faster on a spreadsheet."

Our first listing shows how you can make logical comparisons with a spreadsheet. The cells in Column A contain expressions that evaluate as true or false. You are looking at the result in the listing. The first expression was entered into the cell by typing, `+A3=A4`. A quick glance at cells A3 and A4 confirms the value reported. The next expression in the column was entered by typing `+A3<>A4`. Note that the Equal sign and the greater-than less-than operator

```
ØØD1        (* call pixscroll with a pathname :  pixscroll("/d1/PIX") *)
Ø1ØD        (* requires Tim Harris' 'd' utility *)
Ø133        (* see the RAINBOW Guide to OS9 book for the source *)
Ø169
Ø16A
Ø16B        PARAM pixdir:STRING[255]
Ø177
Ø178        DIM pixname:STRING[29]
Ø184        DIM cr,char:STRING[1]
Ø194        DIM offset,pixel,dirfile,pixfile,errnum:BYTE
Ø1AB        DIM delay:REAL
Ø1B2        DIM gfxmem:INTEGER
Ø1B9
Ø1BA        cr:=CHR$($ØD)
Ø1C3        offset:=5
Ø1CA
Ø1CB
Ø1CC        ON ERROR GOTO 1ØØ
Ø1D2
Ø1D3 1Ø
Ø1D7        CHD pixdir
Ø1DC        SHELL "d >pixlist.dir"
Ø1EE
Ø1EF        ON ERROR
Ø1F2
Ø1F3        OPEN #dirfile,"pixlist.dir":READ
Ø2Ø9        SEEK #dirfile,Ø
Ø212
Ø213        WHILE NOT(EOF(#dirfile)) DO
Ø21E          pixname:=""
Ø225          LOOP
Ø227            GET #dirfile,char
Ø231          EXITIF char=cr THEN
Ø23E          ENDEXIT
Ø242            pixname:=pixname+char
Ø24E          ENDLOOP
Ø252          IF LEN(pixname)>4 THEN
Ø25F            IF RIGHT$(pixname,4)=".pix" THEN
Ø272              RUN gfx("mode",Ø,5)
Ø284              RUN gfx("gloc",gfxmem)
Ø295              OPEN #pixfile,pixname:READ
Ø2A1              SEEK #pixfile,offset
Ø2AB              WHILE NOT(EOF(#pixfile)) DO
Ø2B6                GET #pixfile,pixel
Ø2CØ                POKE gfxmem,pixel
Ø2C9                gfxmem:=gfxmem+1
Ø2D4              ENDWHILE
Ø2D8              CLOSE #pixfile
Ø2DE              FOR delay=1 TO 7ØØØ
Ø2F1              NEXT delay
Ø2FC            ENDIF
Ø2FE          ENDIF
Ø3ØØ        ENDWHILE
Ø3Ø4        CLOSE #dirfile
Ø3ØA        RUN gfx("quit")
Ø316        KILL "gfx"
Ø31D        DELETE "pixlist.dir"
Ø32C        END
Ø32E
Ø32F 1ØØ
Ø333        errnum:=ERR
Ø339        IF errnum=218 THEN
Ø345          DELETE "pixlist.dir"
Ø354          GOTO 1Ø
Ø358        ELSE
Ø35C          PRINT "OS9 error #"; errnum
Ø36F          BYE
Ø371        ENDIF
Ø373

*******************************************
```

Listing 4:

```
chd /dØ/cmds              =d7
load dump                 . .+11f
debug                     =c4
ldump                     =fd
                          =57
```

in Column B are merely labels — they are simple ASCII text. To enter the equal sign, merely point to the cell and then type =, etc.

Speaking of ASCII text, *DynaCalc* has a feature that beats the socks off *Multiplan*. Notice the message telling about the operators. It is stored in a number of different cells all the way across the screen. On *Multiplan* you had to type the first cell, move to the second and pick up typing the word in the middle, then move again, until reaching the end of the line. With *DynaCalc*, point to the cell where you want to start your message. Then, type /A for attributes, followed by /L for label. Now, type the entire message at one time. *DynaCalc* will automatically enter the proper characters in each cell.

The spreadsheet that demonstrates *DynaCalc's* @IF function is amazing. Look in the first row of data. The expression that generated the $65.00 in Column H is @IF(D4=H2,A4,A20. Let's study this.

D4 contains the label "util." Now look at Cell H2. It also contains the label util. Since the expression D4=H2 evaluates as true, *DynaCalc* stored the value of Cell A4 — or 65.00 in Cell H4. If the expression had been false, *DynaCalc* would have stored the value of Cell A20 in Cell H4. When we look closely at Cell A20 we see that it contains a label — in this case, a blank space.

To help you fill out your spreadsheet, Cell E4 contains the expression @IF (D4=E2,A4,A20. In English, it reads "If Cell E2, which contains the type of expenditure, contains the word "food," we will store the value of the money spent, Cell A4 here. Otherwise, we will store a blank space." Since Cell D4 contained util and util is not equal to "food," *DynaCalc* stored a blank space in Cell E4. By the way, Cell A19 contains the expression @SUM(A3 . . . A18. The other totals in this row contain similar expressions with only the column address changed. I'll leave it up to you to write the expression in Cell I19 that checks to see if the total in A19 is the same as the sum of the totals of the individual categories.

Our third listing shows how to set up a spreadsheet to keep track of your checkbook. All of the columns except the last contain values. The first three are labels. The next to last two contain numerical values that someone has entered. The calculation takes place in the last column. For example, Cell F4 contains the expression F3+E4–D4, Cell F5 contains F4+E5-D5, etc. In

```
 . .+1Ød
 =ØØ
 . .+17
 =ØØ
 . .+Ø5
```

```
q
del dump
save dump
unlink dump
chd /dØ
```

****

Listing 5:

```
*************************************************
*
* DIR  -  COPYRIGHT (c) 1985 by S. B. GOLDBERG
*
* Replacement Dir utility for any 8Ø column screen.
*
* Syntax is the same as standard OS-9 Dir utility:
*
*    Dir [e] [x] [directoryname or pathname]
*
         ifpl
         use   /dØ/defs/os9defs
         endc
*
         mod   len,name,prgrm+objct,reent+l,entry,dsiz
*
count    rmb   1   column counter
dpath    rmb   1   directory path number
dskpath  rmb   1   disk path number
mode     rmb   1   execution mode flag
dire     rmb   1   dir e flag
display  rmb   3   line display for dir e
owner    rmb   1   owner number digit
nowtime  rmb   4   owner number digit; start of time display
year     rmb   16   last mod. date
attribs  rmb   11   attributes
sector   rmb   9   starting sector
size     rmb   7   file size
szend    rmb   3
dentry   rmb   3Ø   filename
mssectr  rmb   2   ms bytes descriptor sector number
lssectr  rmb   1   ls byte descriptor sector number
bytes    rmb   1   ls byte descriptor sector position
stuff    rmb   2   file descriptor data
who      rmb   1   ls byte of owner number
yr       rmb   6   last modified date
sze      rmb   4   file size
         rmb   2ØØ   stack
         rmb   2ØØ   parameters
dsiz     equ
*
*
name     fcs   /dir/
         fcc   /(c) 1985 S.B.GOLDBERG/
*******************************
*
* INITIALIZE
*
entry    pshs  x   save parameter pointer
         clr   dire   clear dir e flag
         clr   mode   clear exec. flag
         clr   bytes   clear ls byte descriptor sector position
         lda   #5   column count
         sta   count   save it
*******************************
*
* CHECK FOR OPTIONS ( X & E )
*
         bra   testopt   check for options e x
optloop  lda   ,x+   get option char.
```

English, the sentence would read "Our balance is equal to the balance in the row above plus the value of any deposits made minus the balance of any checks written." The author used *DynaCalc's* replicate command in relative mode to enter the expressions in cells F5 through F20.

The final listing shows how easy it is to calculate Net Present Value using *DynaCalc*. Everything in Column A is a label. Everything except the last cell in Column B contains a value entered by the operator. The calculation is done by the expression typed in the last cell in the column, C5+@NPV(C3/100,C7 . . . C11. Now, pull out one of your BASIC text books and look at the BASIC program that's needed to type in when you want to make the same calculations. Which is easier?

Thunder RAM Breaks the Speed Barrier

The performance of my CoCo has literally skyrocketed. I installed a *Thunder RAM* 256K memory upgrade board from Bob Rosen at Spectrum Projects, (93-15 86th Drive, P.O. Box 21272, Woodhaven, NY 11421, phone 718-441-2807) in my 'D' model CoCo. Frankly, I am amazed.

Thunder RAM is very easy to install and the entire project took less than a half hour. The software is also a snap to install, thanks to procedure files written by Brian Lantz. The 256K of memory available from the *Thunder RAM* gave Lantz enough room to emulate a single-sided, 40-track floppy disk. As Brian says in his documentation, "Anything you can do with a floppy disk, you can do with *RamDisk*. Instead of using /D0, use the name of the *RamDisk's* device descriptor, /R0."

Lantz's *RamDisk* driver appears to be compatible with all OS-9 utility commands and application programs. Here's some food for thought. You can even backup between your floppy disks and the *RamDisk* in *Thunder RAM*. We'll show what this means in time saved when we look at some timings.

Speed is the key word here. You can read 5,000 sectors from a *RamDisk*, or write them to it in a single second. Of course, you can't force 5,000 sectors into the 256K *Thunder RAM*. This speed will make floppies seem painfully slow and even make a hard disk look like it is gummed up.

There's only one catch. If the power goes out, you lose. The contents of your *RamDisk* will be lost. This means if you are using it to store data, you should

```
         cmpa  #$Ød   done?
         beq   findname   yes, go find dir. name
         cmpa  #$2Ø   space?
         bne   optloop   no, look some more
testopt  ldd   ,x   get option chrs.
         cmpb  #$2Ø   space?
         bhi   optloop   no, look again
         ora   #%ØØ1ØØØØØ   make lower case
         cmpa  #'e   e ?
         beq   sete   yes, set dir e flag
         cmpa  #'x   x ?
         bne   optloop   no, look some more
         lda   #exec.   execute mode
         sta   mode   save it
         bra   clear   clear option
sete     inc   dire   set dir e flag
clear    clr   ,x+   clear option
         bra   optloop   look again
********************************
*
* FIND NAME ON COMMAND LINE
*
findname puls  x   retrieve param. pointer
nameloop lda   ,x+   get param. char.
         cmpa  #$Ød   directory name?
         beq   nodir   no, use current dir.
         cmpa  #$2Ø   directory name?'
         bls   nameloop   no, look again
         bra   chngdir   yes, go ahead
nodir    leax  head,pcr   default, current dir.
********************************
*
* CHANGE DIRECTORY IF NEW DEVICE
*
chngdir  lda   ,-x   get first char. of pathname
         pshs  x   save address of pathname
         cmpa  #'/   new disk drive?
         bne   time   no, go ahead
         lda   #read.   maybe, better
         os9   i$chgdir   change the directory
         lbcs  out   exit with error
********************************
*
* SET UP TIME DISPLAY
*
time     leay  timeform,pcr   time display format
         leax  nowtime,u   data address for time display
timeloop lda   ,y+   get display char.
         sta   ,x+   put it in data area
         bne   timeloop   not done, repeat
         os9   f$time   get current time
         lbcs  out   exit with error
         leax  3,x   hour byte
         leay  year,u   time display address
         lbsr  setdate   set time display
********************************
*
* OPEN DIRECTORY
*
         ldx   ,s   get pathlist pointer
         lda   #read.+dir.   read directory mode
         adda  mode   add exec. mode if 'Dir x'
         os9   i$open   open directory
         lbcs  out   exit with error
         sta   dpath   save dir. path number
         tfr   x,d   end of dir. name
         subd  ,s   subtract start of name
         pshs  d   save pathname length
         tst   dire   dir e?
         beq   header   no, put header on screen
********************************
```

copy files to a real floppy disk regularly.

Here's something else you can do with *Thunder RAM*. After using BACKUP to place a working disk in device /R0, put your boot floppy disk in Drive /D0 and hit the Color Computer's Reset button. After booting OS-9 again, type CHX /RØ/CMDS and CHD /RØ. Now, do a DIR. Your *RamDisk* will stay intact, even though you reboot with the Reset button. It's another feature that comes in handy and saves you a lot of time.

Ways You Can Use *RamDisk*

You can use a *RamDisk* in several different ways. One of the handiest is to make up a 40-track, single-sided floppy disk that contains the programs and system files you use all the time. As soon as you boot OS-9, format your *RamDisk* (it takes about five seconds) then use the BACKUP utility command to back up your floppy in the *RamDisk*. Next, use the CHX command to change your working execution directory to /R0/CMDS. You won't believe it when you see the almost instant response to your commands on the screen. You won't know how you ever lived without it.

The other way to use the *RamDisk* is to hold temporary files generated by high level language compilers like OS-9 C and OS-9 PASCAL. Mike Bailey (Unified Software, 525 South Chestnut, Cameron, MO 64429), author of *The Last Word*, a mouse-driven editor for OS-9, is using his *Thunder RAM* in this manner and reports he is able to compile a C program three to four times faster than before.

To use the *RamDisk* like this, name the *RamDisk* as the current data directory with the CHD command and read the file you are compiling from a floppy disk. Or, copy the file you are compiling into the RAM disk and then compile it. In any case, it is always a good idea to edit your programs on the floppy disk. This gives you a way to start over if a program crashes or there is a power failure that wipes out the *RamDisk*. If you edit your programs and save them in the *RamDisk*, make sure you save them to a real floppy frequently.

I promised we would talk speed and give some timing comparisons. It takes 56 seconds to back up and verify a 40-track, single-sided floppy disk to the *RamDisk* in *Thunder RAM*. When I did a DSAVE of the standard Radio Shack OS-9 CMDS directory from my original Tandy disk to the *RamDisk*, it took two minutes and 29 seconds. I then

```
* OPEN ENTIRE DISK FOR DIR E
*
        leax  at,pcr   address of @ (entire disk)
        lda   #read.   read mode
        os9   i$open   open disk
        lbcs  out   exit with error
        sta   dskpath   save disk path number
******************************
*
* PUT HEADER ON SCREEN
*
header  leax  head,pcr   start of header
        ldy   #15   length
        lbsr  print2   put start of header on screen
        puls  y   pathlist length
        puls  x   pathlist address
        bsr   print2   put on screen
        leax  nowtime,u   time display
        bsr   print1   put time on screen
        tst   dire   dir e?
        beq   read   no, get first filename
        leax  subhead,pcr   yes, address of column headings
        bsr   print1   put on screen
******************************
*
* READ AND DISPLAY FILE NAMES
*
read    leax  dentry,u   address for filename
        lda   dpath   directory path number
        ldy   #29   maximum filename length
        os9   i$read   get filename
        bcs   done   branch with error
        leax  mssectr,u   address for descriptor sector number
        ldy   #3   three bytes
        os9   i$read   get desciptor sector number
        bcs   done   branch on error
        lda   dentry,u   first filename character
        anda  #%Ø1111111   clear ms bit
        cmpa  #'A   valid filename?
        blo   read   no, get next name
        clrb   yes, zero character counter
        leax  dentry,u   address of filename
loop    incb   count character
        lda   ,x+   last character?
        bpl   loop   no, get another
        anda  #%Ø1111111   yes, clear ms bit
        sta   -1,x   return to filename
        lda   #$Ød   carriage return
        sta   ,x   terminate filename
        tst   dire   dir e?
        bne   doitall   yes, do full directory listing
        clra   no
        tfr   d,y   length of filename
        leax  dentry,u   address of filename
countit dec   count   column 5?
        beq   endline   yes, print with carriage return
        subb  #16   longer than 15 characters?
        bpl   countit   yes, take another column
        bsr   print2   print filename
        leax  space,pcr   address of fill spaces
        clra
        negb   make positive
        tfr   d,y   number of spaces to fill column
        bsr   print2   fill entire column
        bra   read   get next filename
endline bsr   print1   end of line, print with carriage return
        lda   #5   new column count
        sta   count   save it
        bra   read   get next filename
******************************
*
```

did a DSAVE of the same directory on a 40-track floppy disk in Drive /D0 to another in Drive /D1. This took six minutes and 37 seconds. By comparison, when I DSAVEd the same directory in *RamDisk* /R0 to a floppy disk in Drive /D1, it took four minutes and 58 seconds.

Other *RamDisks* on the Horizon

To be fair, we must mention several other hardware and *RamDisk* software now hitting the CoCo OS-9 market. D. P. Johnson, who revolutionized OS-9 on the Color Computer by writing *SDISK*, has introduced a new RAM disk cartridge that adds 512K bytes of memory to CoCo. Called "CCRD," it occupies four bytes of the 6809's memory map. Two of the bytes are used as a sector register to set the logical number of the 256 byte sector that will be displayed in a two-byte data window. Two CCRD cartridges can be plugged into the Color Computer extension interface at the same time to give one megabyte of *RamDisk* storage. Johnson sells an OS-9 device driver named "CCRDV" for use with the CCRD.

Kevin Darling of Darling Software, 3081 Stonybrook Drive, Raleigh, NC 27604, has written a *RamDisk* driver for the DSL 128K memory upgrade board. It's used in the same manner you use Lantz's drivers with the *Thunder RAM*, except there is only 128K of memory, so it can't emulate a 40-track floppy disk.

Another 256K RAM memory upgrade board named the "Banker" is being marketed by Jesse Jackson and Ray Rowe, of J & R Electronics, P. O. Box 2572, Columbia, MD 21045, phone (301) 987-0578. They made a presentation to the Northern Virginia Color Computer Club in Manassas,but I was unable to attend the meeting.

The GIMIX Single Board 68020 Entry

While we're on the subject of new hardware and software, we've got two more items that should perk your interest. First, on the software front, *K-BASIC* from Lloyd I/O has been updated. It will now compile *TSC BASIC*, *XBASIC* and *XPC* source code files. This means you can run any *TSC XBASIC* software under OS-9. You need only transfer the source code to OS-9, compile it, assemble it and find your favorite FLEX application software running under OS-9.

On the hardware front, speed is again the keyword. Richard Don at GIMIX, 1337 West 37th Place, Chicago, IL

```
 * ERROR CHECK AND TERMINATION
 *
 done      cmpb  #e$eof   end of file?
           bne   out   no, quit with error
           leax  cr,pcr   yes, carriage return
           bsr   printl   print it
           clrb   clear error
 out       os9   f$exit   quit
 ******************************
 *
 * SCREEN PRINT SUBROUTINE
 *
 printl    ldy   #255   maximum line length
 print2    lda   #1   standard output path
           os9   i$writln   write to screen
           bcs   out   exit with error
             s   return
 ******************************
 *
 * SET DISPLAY FOR DIR E
 *
doitall  leax  form,pcr   address of display line
         leay  display,u   data address for display
formloop lda   ,x+   get character
         beq   sectr   branch if done
         sta   ,y+   not done, save it in data area
         bra   formloop   get next character
******************************
*
* SECTOR DISPLAY
*
sectr    leax  mssectr,u   ms byte of sector number
         leay  sector,u   address of sector display
         ldb   #3   three bytes
sctrloop lda   ,x+   get byte
         lbsr hex   convert to hex
         decb   done?
         bne   sctrloop   no, get another
         leax  sector,u   yes, start,of number display
         bsr   scratch   clear leading zeros
******************************
*
* GET FILE DESCRIPTOR,DATA
*
         ldx   mssectr   most sig. bytes of sector position
         pshs  u   save U register
         ldu   lssectr   least sig. bytes of sector position
         lda   dskpath   disk path number
         os9   i$seek   find file descriptor sector
         bcs   out   exit with error
         puls  u   retrieve U register
         leax  stuff,u   address for file descriptor data
         ldy   #13   13 bytes
         os9   i$read   get descriptor data from disk
         bcs   out   exit with error
******************************
*
* ATTRIBUTE DISPLAY
*
         leay  attribs,u   attribute display address
         ldd   #$2dØ8   a="-" for blanking; b=8 counter
attrloop lsl   ,x   attr set?
         bcs   nextattr   yes, check next
         sta   ,y   no, blank it out
nextattr leay  1,y   next attr
         decb   done?
         bne   attrloop   no, get next
******************************
*
* OWNER NUMBER DISPLAY
*
```

60609, phone (312) 927-5510, has announced the GMX Micro-20 single-board computer. You'll find this board is a powerful, compact 32-bit computing system featuring two megabytes of memory, serial and parallel I/O ports, a floppy disk controller, a SASI interface for intelligent hard disk controllers and a time of day clock with battery backup. If you're really in a hurry, plug in an optional MC68881 floating-point co-processor for number crunching. It can also be used with an optional network interface that supports up to 32 nodes. Frank Hogg at FHL plans to feature a model of his QT computer that uses this board.

GIMIX ran several benchmark programs on their new board and supplied the results. A program named ***bnchtour*****, which executed in 42 minutes and 19 seconds on the GMX Micro20 board running at 12.5 MHz, took two hours and 28 minutes to run on a MIZAR 68000 running at 10 MHz and three hours and 39 minutes to run on a QT using the UniQuad 68008 board with an eight MHz clock speed.**

Computerware Offers More OS-9 Software

Paul Searby at Computerware, 4403 Manchester Ave., Suite 102, Box 668, Encinitas, CA 92024, phone (619) 436-3512, is another vendor who has made a strong showing in the OS-9 world in the past six months. I received three programs from Paul — a text processor, a new debugger and a Macro assembler. Additionally, he is preparing to issue another OS-9 utility package in the near future. Among other things, it will contain a new Shell that will make your 6809 OS-9 look like 68K OS-9 — wild cards and all.

Computerware's OS-9 *Text Formatter* ranks with the best. If you've used the TSC text processor with the FLEX operating system, you'll be right at home. Also, if you've used the UNIX "nroff" text processor on a minicomputer at work you'll be impressed. Computerware has gone out of the way to make the OS-9 *Text Formatter* as compatible as possible with "nroff." By the way, "nroff" stands for "new run off" in UNIX speak.

When working with a text processor, you create your product in two passes. First, you edit the copy with a text editor, then you list the copy through the text formatting program. When typing the copy, you enter special formatting commands in the file by typing a period followed by a two-letter

```
        lda   who   ls byte of owner number
        leay  owner,u   owner display address
        bsr   dec   binary to decimal
        lda   owner   get first digit
        cmpa  #$2Ø   number?
        beq   fsize   no, continue
        adda  #16   yes, make ASCII
        sta   owner   save it
******************************
*
* FILE SIZE DISPLAY
*
fsize   leax  sze,u   ms byte file size
        leay  size,u   size display address
        ldb   #4   counter
szloop  lda   ,x+   get byte of size
        bsr   hex   convert to hex
        decb  done?
        bne   szloop   no, get next byte
        leax  size,u   yes
        bsr   scratch   clear leading zeros
        cmpb  szend   size=Ø?
        bne   date   no, continue
        sta   szend   make it zero
******************************
*
* LAST MODIFIED DATE DISPLAY
*
date    leay  year,u   address of date display
        leax  yr,u   start of date
        bsr   setdate   set the date of modification
        lda   ,x+   get hour
        bsr   dec   make ASCII
        lda   ,x   get minutes
        bsr   dec   make ASCII
******************************
*
* DISPLAY FILE DATA
*
        leax  display,u   start of display line
        lbsr  printl   put line on screen
        lbra  read   get next filename
******************************
*
* DELETE LEADING ZEROS
*
scratch ldd   #$3Ø2Ø   a="Ø"; b=$2Ø (space)
ziploop cmpa  ,x   check numeral
        bne   home   no, return
        stb   ,x+   yes, make space
        bra   ziploop   try again
******************************
*
* BINARY TO HEXADECIMAL
*
hex     pshs  a   save byte
        lsra  move the
        lsra  most signifcant
        lsra  nibble to the
        lsra  least sig. position
        bsr   hex2   make hex
        puls  a   get byte again
        anda  #%ØØØØ1111   kill most sig. nibble
hex2    adda  #'Ø   make ASCII
        cmpa  #'9   greater than 9?
        bls   save   no, save it
        adda  #7   yes, make letter (A-F)
save    sta   ,y+   save hex char.
home    rts   return
******************************
```

code and optional parameters at the beginning of a line. Later, when listing the copy to the printer via the text formatter, it will read your commands and format your text accordingly. For example, if you have been typing single-spaced copy but want to switch to triple spacing for emphasis, simply type the command `.ms 3` at the very beginning of a line in your copy. All lines following this command will be triple-spaced until the processor reads a single-space, command.

Computerware's *CBug Debugger* is also impressive. Written by Chris Bone, it is a screen oriented debugger that features a built-in disassembler, single step capability, a memory window and access to the OS-9 Shell. *CBug* will run on a standard CoCo 32 character screen, a 51 by 24 Hi-Res screen or an 80 by 24 screen such as those available from PBJ's *Work-Pak II*. Information about the current state of your microprocessor is displayed along the bottom of any of these screens. The rest of the area is used for a memory edit window, and to list breakpoints and disassembled listings.

The *CBug* display of the 6809's condition code register is clever. Seven letters are always displayed — fhinzvc. However, if one of these bits is set, it is displayed as an uppercase character. Otherwise, it will always appear as a lowercase letter.

CBug sets breakpoints and single steps through your program by placing a software interrupt instruction at the address where you want the program to stop. You may disassemble a portion of your code by typing a 'D' followed by the starting address of the code you want to study. Normally, your disassembly will go to the Color Computer screen. However, if you want to send it to a printer or a disk file, type a space followed by a complete pathlist after the address. During the next month we'll try to take a look at Computerware's OS-9 *Macro Assembler*.

This Month's Listings

This month we feature three handy Shell scripts or procedure files, a BASIC09 program that scrolls pictures on the CoCo screen and a C filter for CompuServe users.

Our first contribution comes from Donald L. McGarry of Northport, N.Y. "Since these were developed shortly after I had read your book, I was interested in using Shell scripts to accomplish any and all tasks for practice," he said. "The result is fully automatic. You wind up with new DIR and

```
*
* BINARY TO DECIMAL
*
dec       cmpa  #1Ø   less than 1Ø?
          blo   unit   yes, set units
          inc   ,y   add 1 to 1Ø column
          suba  #1Ø   subtract 1Ø
          bra   dec   look again
unit      leay  1,y   less than 1Ø
          adda  #'Ø   make ASCII
          sta   ,y+   save it
          rts   return
*******************************
*
* DATE DISPLAY
*
setdate  ldb   #3   3 bytes
dtloop   lda   ,x+   get byte
         bsr   dec   make decimal ASCII
         leay  1,y   skip separator
        decb   done?
        bne   dtloop   no, do again
        rts   yes, return
*
*
subhead  fcb   $Øa
         fcc   / Owner  Last Modified  Attributes  Sector  Bytecount  Name/
         fcb   $Øa
         fcc   /----------------------------------------------------------/
cr       fcb   $Ød
form     fcc   "    Ø   ØØ/ØØ/ØØ ØØØØ   dsewrewr"
space    fcc   /                        /
         fcb   Ø
dot      fcc   /./
head     fcb   $Øa
         fcc   / Directory of /
at       fcc   /@/
timeform fcc   "    ØØ:ØØ:ØØ"
         fdb   $ØdØØ
         emod
len      equ   *
         end

*************************************************************************
```

Listing 6:

```
#include <stdio.h>
main()
{
   int c,i,j,last;
  last=32;
  while((c=getchar()) != EOF) {
   i = '\Ø';
   if( c > 3Ø )
    if( (c == '('  || c == '*' ) && ( last == '\n'  || last == '\r' ) ) {
       i=getchar();
       if (( c == '(' && i == 'U' ) || (c == '*' && i == ' '))
       {
         while((c=getchar()) != '\n' )
           ;
       }
      else
        {
          putchar(c);
          putchar(i);
        }
     }
    else
      putchar(c);
   if( c==1Ø || c==13)
       if(last != 1Ø && last != 13)
         putchar(c);
   last = c;
  }
}
*************************************************************************
```

MDIR commands plus a renamed copy of the original command. It's a no-lose situation, if something goes wrong, nothing is permanently altered."

After running the procedure file named *wide.dir*, you will have a new copy of DIR in your current execution directory, /D0/CMDS. You will also see another file named *Dir32* in /D0/CMDS. The listing "wide.mdir" works in the same manner.

While on the subject of 80-column displays, here's a short Shell script that modifies the DUMP utility command for 80-column output without the use of the "-L" option. It was contributed by Stephen B. Goldberg of 695 Plainview Road, Bethpage, NY 11714.

Whenever possible, we try to show several different ways to accomplish the same task. This month, Stephen Goldberg contributed an assembly language listing of a new DIR command that prints in 80 columns. He said that if you don't want to take the time to type in this listing, send him a standard Tandy OS-9 formatted disk, a self-addressed, stamped mailer and $3 and he will return it to you with both the source and object code.

Goldberg's DIR uses the same syntax as the original utility and spreads the standard directory over five columns across the full width of the screen. The DIR E display is the same as that shown on Page 23 of the OS-9 commands manual. Goldberg writes, "Since we have only 87 terminals connected to our CoCo (just kidding), I didn't implement calculation of all possible user numbers. Only users between zero and 99 will be reported correctly in the owner column of the display. If you run more than 100 users on your CoCo OS-9 system, you will want to rewrite the owner number routine." Enjoy!

Mark W. Smith contributed some more of his work this month and we're featuring his BASIC09 procedure named *pixscroll*. In order to use *pixscroll*, first copy your Radio Shack pictures to OS-9. Smith uses FHL's *O-PAK XCOPY* tool for the job, but there are now several other utilities that can be used. Thanks Mark, *pixscroll* is fun.

Our last contribution this month comes from John Carter of Smyrna, Ga., who has also shared his work in earlier columns. He writes, "I frequently do a scan of the CoCo and OS-9 SIGs on CompuServe, then read just the messages of interest. I then print and file these messages in a binder. I needed a routine to delete the junk lines from the output. I wrote *cis.c* to do the job."

NOW! is the time to subscribe to Australian RAINBOW

Copies of back issues can be obtained, subject to the availability of stocks, by using this order form and marking clearly which issues you require to be sent to you.
Each issue costs $4.50 including postage and packing. Please enclose your cheque / money order made payable to: Australian Rainbow Magazine, PO Box 1742, Southport, 4215.

RATES

AUSTRALIAN CoCo / MiCo / softgold		AUSTRALIAN RAINBOW
□ $3.45	Latest per copy	$4.50 □
□ $19	6 months	$24.75 □
□ $31	12 months	$39.95 □

BACK ISSUES

MiCo —
First Issue Oct '83
□
$3/copy

CoCo / MiCo / softgold
First Issue Aug '84
□
$3.45 / copy

AUSTRALIAN RAINBOW		
Feb '85-Apr '85	□	$3.95
Aug '84-Jan '85	□	$3.25
to July '84	□	$3.00

MiCoOz
(MiCo on Tape) $8 ea □ First Issue Dec '83

DISKS & TAPES

CoCoOz
(Aust CoCo on Tape)
□ Latest $8
□ 6 months $42
□ 12 months $75
First Issue Mar '83
or Debit my Credit Card Monthly □

Blank tapes 12 for $18 or $1.70 ea □
Cassette Cases, 10 for $5 □
Disks — $3.50 ea □ 10 for $29.99 □

Rainbow on Tape (program listings) $12 for month of □
Please note that RAINBOW on TAPE is issued irregularly
or Annually $144 □ or Debit my Credit Card Monthly □

BOOKS

Byte
Elementary $5.95 □

Help
Medium $9.95 □

Facts
Advanced $11.95 □

MiCo Help (MC-10 computer).
Medium $9.95 □

PLUS

□ Best of CoCoOz # 1 (Education)
□ Best of CoCoOz # 2 (Games)
□ Part 1 for 16K ECB
□ Part 2 for 32K ECB (Some 16k CB on P1)
TAPE $ 10.00, DISK $ 21.95

BULLETIN BOARD

Let your computer talk to ours!
CoCoLink —
Annual Sub $29 □

□□□□□□□□□□□□□□
□ VISA BANKCARD □
MASTERCARD □

________________ Signature

□ CASH
□ CHEQUE
□ MONEY ORDER

Authorised
Amount $ ________

BLOCK CAPITALS PLEASE

If you already subscribe to either Australian Rainbow or Australian CoCo please place Subscription No
□□□□□□□

Complete the section below with one letter, figure or space per square

FIRST NAME SECOND NAME
□□□□□□□□□□□□□□□□□□□□

Address
□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□
□□□□□□□□□□□□ PC □□□

Telephone Number STD Code □□□ Local Number □□□□□□□□□□□

New Subscription □ Renewal □

GOLDSOFT

Hardware & Software for your TANDY computer.

HARDWARE

The CoCoConnection: Connect your CoCo to the real world and control robots, models, experiments, burglar alarms, water reticulation systems — most electrical things. Features two MC 6821 PIAs; provides four porgrammable ports; each port provides eight lines, which can be programmed as an input or output; comes complete with tutorial documentation and software; supplied with LED demonstration unit; switchable memory addressing allows use with disk controller or other modules via a multipack interface; plugs into Cartridge Slot or Multipack, uses gold plate connectors; a MUST for the hardware designer and debugger!		$195.00
Video-Amp: Connects simply to your CoCo to drive a Colour or Mono monitor.	With instructions With instructions and sound	$25.00 $35.00

SOFTWARE

CoCoOz: The programs you see listed in Australian CoCo Magazine are available on CoCoOz! No laborious typing — just (C)LOAD and Go!	Each Tape Subscription, 6 months 12 months **NEW** for 1986 ONLY Each DISK Subscription on disk, 12 months	$8.00 $42.00 $75.00 $10.95 $102.50
Rainbow on Tape: Australian. The programs you see listed in Australian Rainbow Magazine are available on tape. A boon if you don't understand the language! American. We also supply the programs found in American Rainbow on tape. Please specify either Australian or American.	Each Tape Subscription, 12 months **NEW** for 1986 ONLY Each DISK Subscription on disk, 12 months	$12.00 $144.00 $15.00 $172.00
MiCoOz: The programs in the MiCo section of Australian CoCo Magazine. (For MC 10 computers only) Back issues of CoCoOz and MiCoOz are always available on tape only.	Each Tape Subscription, 12 months	$8.00 $75.00
To be released soon: **GOLDDISK 1000** — programs from 'softgold' for your Tandy 1000 on disk, and, **GOLDDISK 4** — programs from 'softgold' for your Tandy Model 4!		
The Best of CoCoOz: 1 . . . EDUCATION programs. Fourteen programs for the teacher or parent. 2 . . . Part 1 . . . 16K GAMES. (Mainly ECB). Sixteen programs to keep you on your toes! Part 2 . . . 32K GAMES. Adventures, simulations and arcade games for everyone! 3 . . . UTILITIES. Programs to make your use of the computer easier. Spoolers, Reverse Video, Disk utilities and more. 4 . . . BUSINESS programs. Invoicing, Accounts, Creditors and more.	Tape Disk 3 on Disk Any two tapes	$10.00 $21.95 $16.00 $17.95
CoCoLink: CoCoLink is our Bulletin Board which you can access with any computer if you have a 300 baud modem and a suitable terminal program. There is a free visitor's facility, alternatively membership entitles you to greater access of the many files available. We can also be contacted through Minerva (OTC) and Viatel (Telecom).	Subscription to CoCoLink, 12 months	$29.00
Magazines: Australian Rainbow Magazine — THE magazine for advanced CoCo users! Australian CoCo Magazine — THE magazine for the new user of a Tandy computer. Also suits owners of CoCos, MC 10s, Tandy 1000s, 100s, 200s & 2000s.		
Back Issues: Australian Rainbow Magazine. (Dec '81 to now.) Australian CoCo Magazine. (Aug '84 to now.) CoCoBug Magazine. For CoCo — usually 8 programs in each magazine. (Sep '84 to Oct '85) Australian MiCo Magazine. For Tandy MC 10 computers. (Dec '83 to Jul '84) Australian GoCo Magazine. For Tandy Model 100 users. (Jul '83 to Jul '84) See Subscription Page for further details.		
Books: HELP: A quick reference guide for CoCo users. FACTS: THE reference for CoCo machine language programmers. MiCo HELP: A quick reference for owners of MC 10 computers.		$9.95 $11.95 $9.95
Othello: by Darryl Berry The board game for your CoCo.	Tape 16K ECB	$15.95
Say the Wordz: by Oz Wiz & Pixel Software Two curriculum based speller programs for your Tandy Speech/Sound Pack.	Tape 32K ECB	$39.95
Bric a Brac: Blank tapes . . . 12 for $18.00 or $1.70 each. Cassette cases . . . 15 for $5.00. Disks . . . (they work!) $3.50 each or $28.95 per box of 10.		

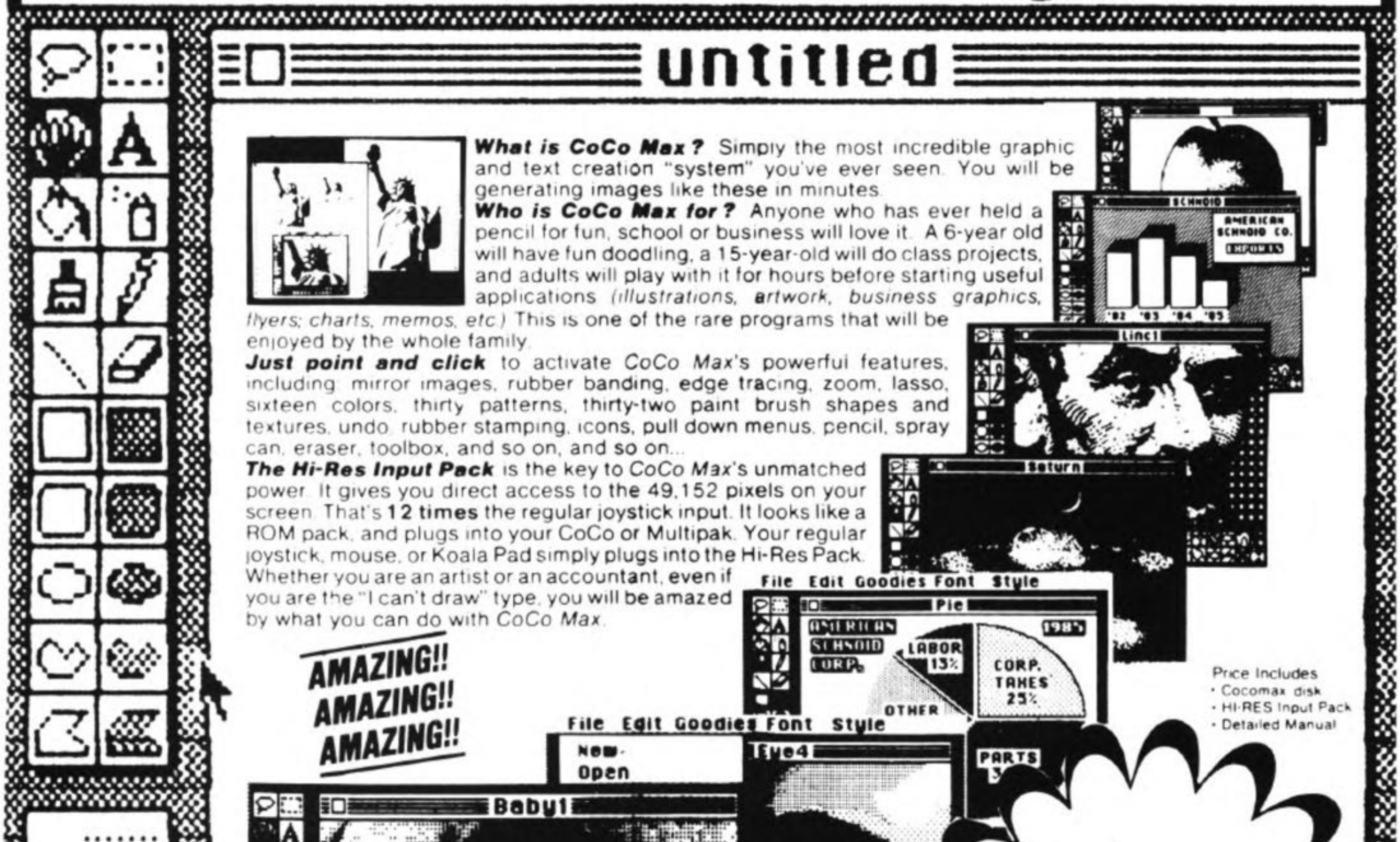
HOW TO ORDER

Option 1: Use the subscription form in this magazine.
Option 2: Phone and have ready your Bankcard, Mastercard or Visa number.
Option 3: Leave an order on Viatel, Minerva or CoColink, but be sure to include your Name, Address, Phone Number, Credit Card Number and a clear indication of what you require, plus the amount of money you are authorising us to bill you.

NOW AVAILABLE
ON THE
QUICKSHIP PROGRAM
from your
TANDY STORE
R90-9715

CoCo Max®

COLORWARE



What is CoCo Max? Simply the most incredible graphic and text creation "system" you've ever seen. You will be generating images like these in minutes.

Who is CoCo Max for? Anyone who has ever held a pencil for fun, school or business will love it. A 6-year old will have fun doodling, a 15-year-old will do class projects, and adults will play with it for hours before starting useful applications *(illustrations, artwork, business graphics, flyers; charts, memos, etc.)* This is one of the rare programs that will be enjoyed by the whole family.

Just point and click to activate *CoCo Max*'s powerful features, including: mirror images, rubber banding, edge tracing, zoom, lasso, sixteen colors, thirty patterns, thirty-two paint brush shapes and textures, undo, rubber stamping, icons, pull down menus, pencil, spray can, eraser, toolbox, and so on, and so on...

The Hi-Res Input Pack is the key to *CoCo Max*'s unmatched power. It gives you direct access to the 49,152 pixels on your screen. That's **12 times** the regular joystick input. It looks like a ROM pack, and plugs into your CoCo or Multipak. Your regular joystick, mouse, or Koala Pad simply plugs into the Hi-Res Pack. Whether you are an artist or an accountant, even if you are the "I can't draw" type, you will be amazed by what you can do with *CoCo Max*.

AMAZING!!
AMAZING!!
AMAZING!!

Price Includes
- Cocomax disk
- HI-RES Input Pack
- Detailed Manual

$99⁹⁵

Disk System: a Multipak or 'Y Adaptor' is needed to plug the disk pack and the HI-RES pack.

SYSTEM REQUIRED:
- Any 64K COCO.
- A standard Joystick, Mouse or Koala Pad.

All these pictures are **unretouched** screen photos or printouts (on an Epson RX-80)

CFM

COMPUTERWARE FOR MICROS.

bankcard welcome here

VISA

Peter & Jillian Collison
11 Grantley Avenue,
Rostrevor S.A. 5073
Phone: (08) 336 6588

AUSTRALIAN RAINBOW MAGAZINE
REGISTERED BY AUSTRALIAN POST —
PUBLICATION NO. QBG 4009
AUSTRALIAN COCO/softgold
REGISTERED BY AUSTRALIAN POST —
PUBLICATION NO. QBG 4007.
PO BOX 1742,
SOUTHPORT. QLD. 4215.

user group CONTACTS

' (Stop between numbers = b.h. else
' a.h.; but, hyphen between = both.)

ADELAIDE	JOHN HAINES	08 278 3560
ADELAIDE NTH	STVN EISENBERG	08 250 6214
ALBURY	RON DUNCAN	060 43 1031
ARMIDALE	TOM STUART	067 72 8162
BAIRNSDALE	COLIN LEHMANN	051 57 1545
BALLARAT	MARK BEVELANDER	053 32 6733
BANKSTOWN	CARL STERN	02 646 3619
BLACKTOWN	KEITH GALLAGHER	02-627-4627
BLACKWATER	ANNIE MEIJER	079.82.6931
BLAXLAND	BRUCE SULLIVAN	047 39 3903
BOWEN	TONY EVANS	077 86 2220
BRASSALL	BOB UNSWORTH	07 201 8659
BRIGHTON	GLENN DAVIES	08 296 7477
BRISBANE:		
EAST	ROB THOMPSON	07 848 5512
SOUTH	PATRICK SIMONIS	07 209 3177
SOUTH WEST	GRAHAM BUTCHER	07 376 3400
WEST	BRIAN DOUGAN	07 300 2072
BROKEN HILL	DEAN PARADICE	080 6701
BUNDABERG	JIM McPHERSON	071 72 8329
CAMDEN	KEVIN WINTERS	046.66.8068
CAMPBELLTOWN	LEO GINLEY	02 605 4572
CANBERRA NTH	JOHN BURGER	062 58 3924
CANBERRA STH	LES THURBON	062 88 9226
CHATSWOOD	BILL O'DONNELL	02 411 3336
' or	MARK ROTHWELL	02 817 4627
CHURCHILL	GEOFF SPOWART	051 22 1389
COLYTON TEENS	DWAYNE MANSON	02 623 5805
COOMA	ROSS PRATT	0648 23 065
COORANBONG	GEORGE SAVAGE	049 77 1054
DALBY	ANDREW B. SIMPSON	074.62.3228
DARWIN	BRENTON PRIOR	089.81.7766
DENILIQUIN	WAYNE PATTERSON	058 81 3014
DUBBO	GRAEME CLARKE	068 89 2095
EMERALD	LEIGH EAMES	059 68 3392
FORBES	JOHANNA VAGG	068 52 2943
FORSTER	GARY BAILEY	065 54 5029
GIPPSLAND STH	PAT KERMODE	056 74 4583
GOLD COAST	GRAHAM MORPHETT	075 51 0015
GOSFORD	PETER SEIFERT	043 32 7874
GOULBURN VALLEY	TONY HILLIS	058 59 2251
GREENACRES	BETTY LITTLE	08 261 4083
GUYRA	MICHAEL J. HARTMANN	067 79 7547
HASTINGS	MICHEAL MONCK	059.86.8288
HERVEY BAY	LESLEY HORWOOD	071 22 4989
HILLS DIST	DENNIS CONROY	02 671 4065
HOBART	BOB DELBOURGO	002 25 3896
HORNSBY	ATHALIE SMART	02 848 8830
IPSWICH	MILTON ROWE	07 281 4059
JUNEE	PAUL MALONEY	069 24 1860
KALGOORLIE	TERRY BURNETT	090.21.5212
KENMORE	GRAHAM BUTCHER	07 376 3400
KINGSTON	WIM DE PUIT	002 29 4950
LEETON	CHRIS NAGEL	069 53 2969
LITHGOW	DAVID BERGER	063 52 2282
LIVERPOOL	LEONIE DUGGAN	02-607-3791
MACKAY	LEN MALONEY	079511333x782
MacQUARIEFIELDS	KIETH ROACH	02 618 2858
MAFFRA	MAX HUCKERBY	051 45 4315
MAITLAND	LYN DAWSON	049 49 8144
MARYBOROUGH	NORM WINN	071 21 6638
MELBOURNE:		
' DANDENONG	DAVID HORROCKS	03 793 5157
' DONCASTER	JUSTIN LIPTON	03 857 5149
' FRANKSTON	BOB HAYTER	03.783.9748
' MELTON	MARIO GERADA	03 743 1323
' RINGWOOD	ANDREW RAWLINGS	03 700 4498
' SUNBURY	JACK SMIT	03.744.1355
MILDURA	SCOTT HOWISON	050 23 6016
MORPHETTVALE	KEN RICHARDS	08 384 4503
MOREE	ALF BATE	067 52 2465
MORWELL	GEORGE FRANCIS	051 34 5175
MT ISA	PAUL BOUCKLEY-SIMONS	077 43 6280
MUDGEE	BRIAN STONE	063-72-1958
MURGON	PETER ANGEL	071 68 1628
NAMBUCCA HDS	WENDY PETERSON	065 68 6723
NARROMINE	GRAEME CLARKE	068 89 2095
NEWCASTLE	LYN DAWSON	049 49 8144
NOWRA	ROY LOPEZ	044 48 7031
ORANGE	STEVE LOVETT	063.62.4025
' or	JIM JAMES	063 62 8625
PARKES	DAVID SMALL	068 62 2682
PENRITH	ALEX SCHOFIELD	047 31 5303
PERTH	IAN MACLEOD	09 448 2136
PORT LINCOLN	JOHN BOARDMAN	086 82 2385
PORT MacQUARIE	RON LALOR	065 83 8223
PORT NOARLUNGA	ROB DALZELL	08 386 1647
PORT PIRIE	KEVÌN GOWAN	086 32 1368
ROCKHAMPTON	KEIRAN SIMPSON	079 28 6162
ROSEVILLE	KEN UZZELL	02 467 1619
SALE	BRYAN McHUGH	051 44 4792
SANDGATE	MARK MIGHELL	07 269 5090
SCARBOROUGH	PETER MAY	07 203 6723
SEACOMBE HTS	GLENN DAVIS	08 296 7477
SHEPPARTON	ROSS FARRAR	058 25 1007
SMYTHESDALE	TONY PATTERSON	053 42 8815
SPRINGWOOD	DAVID SEAMONS	047 51 2107
STURT	MARY DAVIS	08 296 7477
SUTHERLAND	IAN ANNABEL	02 528 3391
SWAN HILL	BARRIE GERRAND	050.32.2838
SYDNEY TEENS	ROD HOSKINSON	02 48 5948
SYDNEY EAST	JACKY COCKINOS	02 344 9111
TAMWORTH	ROBERT WEBB	067 65 7256
TAHMOOR	GARY SYLVESTER	046 81 9318
TARA	STEVEN YOUNGBERRY	
TONGALLA	TONY HILLIS	058 59 2251
TOOWOOMBA	GRAHAM BURGESS	076 30 4254
TOWNSVILLE	JOHN O'CALLAGHAN	077 73 2064
TRARALGON	MORRIS GRADY	051 66 1331
UPPER HUNTER	TERRY GRAVOLIN	065 45 1698
WAGGA WAGGA	CES JENKINSON	069 25 2263
WONTHAGGI	PAT KERMODE	056 74 4583
WYNYARD	ANDREW WYLLIE	004 35 1839
YARRAWONGA	KEN SPONG	057 44 1488
SPECIAL INTEREST GROUPS		
' BUSINESS		
BRIZBIZ	BRIAN BERE-STREETER	07 349 4696
' TDP - TELEWRITER, DYNACALC PROCOLOR		
BRISBANE	GEOFF TOLPUTT	07 44 6084
' OS9 GROUPS		
BANKSTOWN	CARL STERN	02 646 3619
BRISBANE	JACK FRICKER	07 262 8869
COOMA	FRED BISSELING	0648 23263
KALGOORLIE	TERRY BURNETT	090.21.5212
PENRITH	BOB THOMSON	047 30 2468
SYDNEY EAST	JACKY COCKINOS	02.344.9111
SYDNEY NTH	MARK ROTHWELL	02 817 4627
' MiCo GROUPS		
LITHGOW	DAVID BERGER	063 52 2282
PORT LINCOLN	JOHN BOARDMAN	086 82 2385
ROCKHAMPTON	TIM SHANK	079 28 1846
SYDNEY	RAJA VIJAY	02 519 4106
' TANDY 1000 / MS DOS		
BRISBANE	BRIAN DOUGAN	07 30 2072
SYDNEY	ROGER RUTHEN	047.39.3903
' FORTH		
BRISBANE	JOHN POXON	07 208 7820
PORT LINCOLN	JOHN BOARDMAN	086 82 2385
SYDNEY	JOHN REDMOND	02 85 3751

AUSTRALIAN

THIS MONTH A SPECIAL HELP EDITION