

For your TANDY
Color Computer

Registered by Australia Post — Publication. No. QBG 4009

\$3.95

AUSTRALIAN

RAINBOW

April, 1985

No.46

Aussie Rainbow
Welcomes you



Tandy

PARIS RADIO ELECTRONICS

161 BUNNERONG ROAD KINGSFORD N.S.W. 2032
P.O. BOX 380, DARLINGHURST, N.S.W. 2010

- SOFTWARE FOR OS-9**
- Hacker's Kit\$ 38.95
 - Filter Kit#1\$ 39.95
 - SDisk-Boot Fix\$ 59.95
 - XTerm\$ 82.95
 - XWord\$ 99.95
 - XScreen\$ 39.95
 - XMenu\$ TBA
 - OPak\$ 56.00
 - Search & Rescue Utility\$ 56.95
 - Dynamite+ Disassembler\$ 69.00
 - RMS Database Manager\$150.00
 - Flex Operating System\$ 69.00
 - Flex Operating System w/DBasic\$ 99.00

FOR INFORMATION ON THESE
AND MANY OTHERS PHONE

02 344 9111

- BOOKS**
- Basic 09 Tour Guide\$ 29.95
 - Book\$ 29.95
 - Range of American Color Computer Magazines

HARDWARE MODEMS

- Cicada 300\$ Call
- Avtec Multi Modem\$389.00
- Avtec Mini Modem\$189.00
- Word-Pak\$199.00
- Word-Pak II\$249.00
- Monochrome Video Board\$ 35.00,
- Colour Video output Board with Sound\$ 45.00
- C-CBus 6 Port Expansion Bus.....\$219.00
- HJL Keyboard\$138.00
- MC 6809 EP Processor Chip\$ 24.00
- MC 6883 SAM Chip\$ 24.00
- 64K Upgrade Chips\$ 7.80

INFOCENTRE

A BULLETIN BOARD SYSTEM

For Color Computer Users with Modems, we have a Bulletin Board Service called INFOCENTRE.

All users are welcome. However, you must first contact our office for authorization and message codes.

The system includes public domain software, hardware, price lists, book and software reviews, technical updates, a message service, a bulletin board and software retrieval.

Info-Centre also accepts visitors, just type visitor where it asks for your phone number and your name.

02 344 9511



Founder GREG WILSON

**AUSTRALIAN
EDITOR AND PUBLISHER**Graham Morphett
Phone ... 075-51-0015.**CO-EDITOR**

Kevin Mischewski

EDITOR'S ASSISTANT

Sonya Young

**AND GRATEFUL
ASSISTANCE FROM**Brian Dougan
Peggy Annabel
Richard and Judy
Bob Thompson
Paul Humphries
Alex Hartmann
Andrew Simpson
Jim & Sheryl Bentick
Patric Simonis
Annette Morphett
Glen Mischewski**COVER DESIGN**

Jim Bentick

All Programs in this
issue of RAINBOW
are available on
cassette tape**ADVERTISING**ToToAdvertising P.O.Box
5730 Gold Coast Mail
Centre, Bundall, Qld.**DEADLINES**The 7th of the preceding
month of publication**OS-9**Kevin Holmes is the
contact for OS-9
information. He also has
access to OS-9 Software
from the U.S.
His address is:—
39 PEARSON ST.,
NARARA, N.S.W. 2250Printed by:
Australian Rainbow
Magazine
P.O. Box 1742
Southport Qld. 4215**PRINT #-2,**

Many people have suggested to me that computing is an expensive hobby. And I suppose that depending on where you're coming from, one can agree that, yes, it does cost money to get going!

But many of us have other hobbies. A lot of you - in fact, most of you seem to have interests in at least one of the following:

trains (either models or the real thing),
boats (either as a tourist or as an owner),
amateur or citizens band radio,
or photography.

If you have an interest in any of these, then I know why you're poor - and it's not because you spend too much time on your computer!

From my perspective, feeding your computer habit costs far less than any of the above pursuits, but is blamed for a greater proportion of the financial stress!

I contend that per dollar spent, you gain as much, or more, satisfaction from your computer than your other hobby. I don't believe this is any reason to drop other interests, but it does justify a rethink as to where you will spend your leisure dollar.

Look at the pictures.



If you have seen the Macintosh, you'll know that this computer represents the state of the art in screen manipulation.

INDEX

LETTERS	P 4
COCOCONF	Annette Morphett P 5
EDUCATION PAGE	P 6
LET COCO TALK	Fred B. Scerbo P 7
SPACE RACE	Daniel Hamilton P 12
PENGUIN PATROL	Paul Wagorn P 15
WE MEAN BUSINESS	P 18
WATCHING CASH FLOW ...	William S. Bonnell P 18
PERT - PROJECT EVALUATION	Jorge Mir P 24
CoCoLink / Games Competition / Pricing	P 29
A GUIDE TO THE DO'S AND DON'TS of MODEMING	Wayne Day P 30
WEATHER or NOT?	Martin H. Goodman M.D. P 34
ASSEMBLY FILE	Kevin Mischewsky P 41
THE 51 COLUMN SCREEN ...	R. Bartly Betts and programs by
COOKING WITH COCO, Part VI .	Colin J Stearman P 48
LURKEY MANOR	Richard Ramella P 52
GET A BOOST OUT OF OS-9 ..	Charles Robitaille P 57
LISTFILE	Gerry Schechter P 58
MARTHA SAYS	Martha Gritwhistle P 59
SUBSCRIPTION FORM	NEW POSITION
	P 61

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

best prices!

TANDY COMPUTERS & ACCESSORIES

FREE DELIVERY THROUGH AUSTRALIA

90 DAYS WARRANTY

Bankcard & Cheque Orders accepted

BAYNE & TREMBATH
 3 Boneo Rd., Rosebud, Victoria 3940
 Ph. (059) 86 8288. A/h (059) 85 4947
 Hastings 059 791513

KEN UZZELL
 Software



P.O. Box 256, Roseville. NSW. 2069

Phone: (02) 467-1619

- * Largest Range of Software for the TRS-80 Color Computer in Australia.
- * ARCADES-ADVENTURES-GRAPHIC ADVENTURES UTILITIES-APPLICATIONS-OS9-EDUCATIONAL SPEECH SYSTEMS-DIGITIZERS-HARDWARE.
- * Specializing in hard to get software.
- * Prompt mail order service.
- * Happy to answer your general queries.

Send \$2.00 for Comprehensive Catalogue over 40 pages and growing weekly.....

- * ROSEVILLE USER'S GROUP MEETING
 Servicing all your computing needs.

RETURN OF THE
JET-I - \$25

JUST LIKE THE MOVIE CHASE! SPEED THROUGH THE FOREST!
 KNOCK THE STORM TROOPERS INTO THE TREES!
 ...AVAILABLE ONLY FROM US...

...80 COLUMN DISPLAY...

THE DOUBLE 80 PLUS 80 COLUMN DISPLAY PLUGS INTO THE CARTRIDGE CONNECTOR ON YOUR COCO. REQUIRES Y CABLE. SEE DOUBLE DENSITY ADS IN U.S. RAINBOW.

...TERMINAL PROGRAMS...

ULTRA TERM + WORKS WITH THE THREE MOST COMMONLY AVAILABLE 80 COL CARDS. WORD PAK (+m). COLOR POWER II AND DOUBLE 80+ (THE ONE WE SELL). THIS IS FOR DISK USERS ONLY. SO IF YOU HAVE YOUR OWN 80 COL CARD, THIS IS THE TERMINAL FOR YOU.

COLORTERM (TAPE AND DISK) DOES AS ABOVE BUT YOU DON'T NEED THE COL 80 CARD AS IT HAS ITS OWN 42, 51, 64, 85 COL DISPLAY WITH LOWER CASE.

JET I - \$25	80 COL CARD - \$180
ULTRATERM - \$90	COLORTERM (Tape or Disk) - \$80
Y CABLES - \$35	64K UPGRADES (Fitted) - \$90
	128 K UPGRADES (FITTED) - \$260

THE 128K UPGRADE INCLUDES A NEW DISK ROM WITH A DOS THAT IS COMPATABLE WITH MSDOS, BUT INCLUDE NEW FEATURES NOT FOUND IN ANY OTHER DOS, SUCH AS THE FULL SCREEN EDITOR, AS WELL AS THE ORDINARY OS9 BOOT AND RAM ROUTINES. SEE THE RGS ADS IN THE U.S. RAINBOW.

**PLUS ALL THE LATEST GAMES AND OTHER PROGRAMMES
 ...SEND FOR A PRICE LIST...**

ALL PRICES INCLUDE POSTAGE - EXCEPT UPGRADES

J. FRICKER
 QUEENSLAND COLOUR SOFTWARE SUPPLIES
 P.O. BOX 306 - CLAYFIELD 4011
 PHONE (07) 2628869 A/HOURS

The Mac uses a 32 bit CPU with 128K to create this presentation, (and costs a lot of money!).

These pictures are, in fact, from CoCo Max - a brand new product that will be in Australia in the next month or so. CoCoMax fits into the ROM port of CoCo, and the addition of a multipack interface is recommended so you can add your drives and modem.

Max will not be cheap, - but as I often say, anything good costs! Depending on whether the Australian dollar is worth one or two stones, the final cost of CoCoMax could be as low as \$160! It will be distributed in Australia by Computerware for Micros.

Recently it has been my privilege to work on several projects with the staff of Tandy; a change of policy for them. We are breaking new ground in our relationship with Tandy. They have revised several other policies too! The result of these other reversals can be found the centre pages - so this, in a very real way is our "Welcome Tandy" edition.

A whole new range of software has hit the Tandy shops - much of it is really exciting stuff! We will be devoting next issue's reviews to some of this material. Just to wet your appetite, there is a new version of the chess game - if you think you're pretty good at the first version - try this one! Then there is "Frogger" (as in THE Frogger) and there's more.

The Tandy organization has about 350 outlets around Australia, leads the world in its personal computer field and in so doing, employs some very special people. Tandy,

you are indeed welcome in these pages - we thank you for the care that you give to your work, the quality of your product and the quality of your staff.

It seems appropriate, because people have been asking, and because we see this "Tandy Edition" as something of a milestone, that we reveal the policies that guide our endeavours.

They include:

1. To provide you, the reader, with as much quality information for your computer as is economically feasible on a regular monthly basis.

2. To encourage, nurture and assist your study of the computer by publishing appropriate articles and programs and by publishing your work as you grow.

3. To encourage, nurture and assist you to find friendship and assistance in the User Group network. Our magazine is not actually about computers - it is about you!

4. We believe in total equality. Therefore we do not differentiate between male and female, young or old.

5. We try to be honest. We will report honestly matters of interest to our readers, even if such reporting is detrimental to an advertiser.

6. We care. We care enough to be working 7 days a week (mostly!); we care enough to be working 14 hours a day minimum; we care enough because it has become very clear that you all care about us!

7. The magazine should reflect your interests. That is why we love to receive and publish your letters.

RAINBOW BITS

supplier of the latest hardware add-ons for CoCo

CoCo MODEM

AUTOMATIC \$295 STANDARD \$250
(P&P \$5) (Reg. Mail \$10)

- 1 Telecom approved to 300 baud bi-directional modem.
- 2 Plugs into Rom port or expansion pack to work with disk.
(Specify Tandy Controller or other).
- 3 Inbuilt terminal programme on Eprom (no loading).
- 4 Automatic model includes push button phone with auto redial facility.
- 5 Requires no power pack. Works with all models of CoCo.
- 6 Leaves printer port free for continuous down load to printer.

DISK CONTROLLER (TANDY Compatible)

Improved DOS, Gold Plated Contacts.
Has Auto Num., OS9, Flex commands,
inbuilt monitor etc. **\$200**

4-SLOT EXPANSION PACK (MARK II) \$125

Enables disk controller, modem, 80 column card, Eprom programmer, Rom packs etc, to be run/selected at one time. Now supports voice pack modules.

CASE CHANGER

LOWER CASE BOARD made and tested \$70

(simple installation) gives true lower case (with descenders) on screen with reverse video.

VIDEO AMP CARD \$25

(simple installation) for driving video monitor.

PHONE (07) 302072, (07) 3690860

Write to 17 Penley St, The Gap. Q. 4061

RAINBOW BITS

LETTERS

Dear Readers,

Tino, Daniel and I are rather flattered to find that a few people have nominated us for the 'Greg Wilson Award'. We are of course glad that some of you have had fun (and hopefully found some use) in typing the odd contributions we have made to the magazines. However, we feel that the contributors are of secondary importance as compared to the primary effort made by the editors of the magazines. The point is that had WE never existed the magazine would still be around, whereas if Graham Morphet (and the splendid team he has collected around him) had not existed, none of us would of had CoCo, MiCo or Australian Rainbow to read after the sad departure of Greg Wilson.

We therefore believe it is entirely fitting that the award should go to those people who have shouldered the main burden and responsibility of producing - and ever improving - our favourite magazine. Greg Wilson would surely have endorsed this viewpoint.

Anyhow, on our part, we would like to nominate Graham Morphet and Kevin Mischevski for the first 'Greg Wilson Award'.

Bob Delbourgo
Sandy Bay, TAS.

Dear Readers,

Bob conned me into inserting this letter, but we want to make it clear that we are not standing for nomination.

There are so many factors to consider in this, and in any case we don't satisfy the longer term criteria - ie, compared with Bob and family or Dean Hodgson, we are only late-comers!

I thank Bob on behalf of the team, we in turn are flattered by your letter, and even more so Bob, by your friendship.

Graham and the team.

Dear Graham,

I am currently in Long Bay Goal and have a keen interest in computers. One of the programs of the May '84 issue No 35, (the name of the program is 'The Little Runner'), has me intrigued. I am learning programming at the moment and I haven't come across this language. Could you tell me what it is? Also could you recommend any book to help me learn this language.

I do own a computer, it is a 64KECB CoCo. I have been with computers for one year, and have learnt the basic language pretty well and would now like to know is there any other language, I could use with my computer.

I would be grateful for all the help in this matter. I was previously a Rainbow subscriber and had to terminate my subscription after being put in goal. I can't really afford the three dollars a month due to my small goal earnings.

Garry Brownshaw
Matraville, NSW.

Dear Garry,

LITTLE RUNNER was written as DATA statements, which are used by a basic program to load them into the computer as machine code.

It is possible to save this program in machine language, after loading the three sections. To do this you have to find the START ADDRESS, the END ADDRESS, and the EXEC ADDRESS. To get the START ADDRESS .. PRINTPEEK(487)*256+PEEK(488) END ADDRESS .. PRINTPEEK(126)*256+PEEK(127)-1 EXEC ADDRESS .. PRINTPEEK(157)*256+PEEK(158).

After these addresses have been found, and

before EXECuting the program, you CSAVEN 'LITTLER', START ADDRESS, END ADDRESS, EXEC ADDRESS. In future you will be able then to CLOADM the entire program in one go.

Machine Language is the native code of CoCo. Basic is in the computer as a 'foreign' body which really exists only to interpret your instructions to CoCo. If you have access to EDTASM, or even if you just follow Kevin's articles in this magazine, you will begin to understand what machine code is.

An alternative language to tackle might be FORTH, a tutorial upon which occurs most months in Australian CoCo.

We have a number of prisoners around the country supplying articles and making contributions of one type or another, so I hope you won't think that you are alone!

Graham.

Dear Graham,

I have a 16K (non extended) CoCo and I was wondering if there is another magazine you could send me that is better suited to my computer than the Australian Rainbow.

Do you think it would be worth it to send my computer away to have it extended to the 16K ECB. I have been quoted \$100 to have this done.

There is one more thing you might be able to help me with, I wrote my own program to store information on my greyhound and saved it on tape and in the past I have no trouble putting it from tape to computer but now it finds the code word ok, which is 'Dog' and about 3 seconds later it comes up I/O error.

Ray Stanwell
Fairfield, NSW.

Ray,

It is possible that you may find Australian CoCo of greater use than Rainbow. CoCo has the advantage that even if none of the CoCo programs will work on your machine, then at least the MiCo programs should stand a chance.

The real trend is to 64K ECB, and I doubt that our programmers are too worried these days about exceeding 16K. This is particularly so for educational software, where I contend that it is unrealistic to say you are serious and only have 16K.

I really can't say whether it is worth you upgrading your machine. Discuss this with your Meet Contact.

It is likely that your tape has 'crashed'. This magazine in the past, has made a big thing about back ups, and I guess we need to reiterate those statements:

1. Have back ups of every piece of software you own on separate tapes or disks.
2. Keep the second copy in a separate place.
3. Religiously back up - don't do it once and then get slack. There is no need - you have a computer - use it to take the drödgery out of the process.
4. Your files are precious - double back ups are not out of the question.
5. Keep tapes and disks away from magnetic media - this in particular means electrical wires, loudspeakers, motors, and ... CoCo.
6. Save programs you are copying or writing before you RUN them. This applies doubly to anything you type in from Australian CoCo this month!

We'd be interested in seeing your program with

a view to printing it, if you'd care to share it.
Graham.

Dear Graham,

Today I picked up your subscription form from our local Tandy dealer. In his letter to me dated 19th Feb '85, Mr Edward Cox, SUPERVISOR COMPUTER SERVICE of TANDY AUST. LTD, MT DRUITT NSW, recommends your magazine as further reading. Upon my question re the popularity of Bulletin Board Systems (BBS) in Aust. Mr E. Cox advised that you maintain a BBS. I assume that this BBS is the Bulletin Board CoCo Link that you advertise on your subscription form.

My attention was first drawn to BBS by two excellent articles in the Nov '84 edition of the American COMMODORE dedicated magazine COMPUTE'S GAZETTE, which I had borrowed from a neighbour's boy, who has a Commodore 64 computer. I enclose printouts of both articles, in case you wish to Australianise and CoCoise and use them. At the same time they can now serve as common ground for comparison. How is the popularity of BBS here in Australia compared in the USA as described in 'Bulletin Board Fever' and 'The Indispensable Sysop'? Who makes Direct Link Modems that are Telecom approved for the Tandy CoCo?

Naturally I had heard of Computer talking to Computer, but, like everyone else, this information came from Hollywood, where computers can transfer whole libraries of information in a matter of seconds. Apparently the reality looks different. The quality of telephone lines limits the transmission speed. How fast is the transmission speed over telecom lines? How long would it take if I, for example, instead of enclosing a hard copy of 'Bulletin Board Fever' to this letter, would prefer to send you that article by uploading it into your BBS.

I am looking for a not-too-advanced book on Computer Technology, but not on programming. Do you publish reviews?

Looking forward to hearing from you.

K. Peter Wolf
Forster, NSW.

Peter,

BBS are relatively new in Australia, and the articles you sent reflect the slick yankie journalistic approach to the subject which suggests that the writer has little experience of his subject and even less interest in it.

The reality of US BBS is different to that described. In Australia, The Australian Beginning, Infocentre, and the guys in WA from the Perth User's group have as long an experience with BBS as anyone.

The BBS is yet to find a niche in the day to day affairs of man, but of all the current technology, as I have previously said, I believe that the BBS is the item that could make the home computer relevant to the average person.

BBS upload / download times are about equivalent to cassette loading times. So if you time a save of your recent letter, you'll get some idea of the comparison.

We are willing to publish reviews of books, but have not received any for review!

You are fortunate, you have an active and capable user group at Foster, front lined by Gary Bailey. He can assist you further.

Graham.

COCOCONF

Annette Morphet.

Hi there! For those of you who are coming to the conference, and are unfamiliar with the Gold Coast, here's an accomodation guide which should be of assistance.

We suggest you ring, or write, and make your own arrangements - see if you can be sent a pamphlet showing the motel / caravan parks and detailing their facilities. Some places I've seen and others I've only picked out of the telephone directory. So here goes:

MOTELS:

1. Beaconlea Apartments, 316 Marine Parade, Labrador, 4125. (4KM away from conference site). Phone 075-32-9919.

Multi-story apartment block, overlooking the Broadwater. Truly delightful, for those who can afford a little extra. Very spacious, with tasteful decor, breathtaking views completely self-contained and includes washing machine, dryer, kitchen and two bedrooms. Tranquil setting, security parking, games room, half-size tennis court. Accomodate 4 people.

Tariff: \$40 a double (includes linen charge)
min 3 nights
\$25 each extra person p.w.

Awarded my highest recommendation (I love it!)

2. Sundale Motel, 20 Queen Street, Southport, 4215, (2KMs from conference site). Phone 075-32-2111. No cooking facilities or restaurant.

Tariff: \$36 double
\$30 single
\$6 each extra person
\$3 for continental breakfast

3. South Surfers Motor Inn, 2884 Gold Coast Highway, Surfers Paradise (5KMs from conference) Phone 075-38-6955. Undecover parking, serviced and self contained units, air conditioned, BBQ, games room, close to beach.

Tariff: \$28 double
\$25 single
\$5 for each extra person
\$35 for 4 people in a self-contained unit.

4. Crystal Waters Motel, Marine Parade, Labrador, 4215 (4KMs from conference) Phone 075-31-1084.

Quiet street, opposite park and beach. All self-contained units with 2 bedrooms and kitchenette. Neat, comfortable, middle-of-the-road accomodation.

Tariff: \$30 double
\$20 single
\$40 family

5. San Martino Motel, 125 Frank Street, (Gold Coast Highway), Labrador, 4215. (3KMs from site) Phone 075-31-3498.

Some self-contained units with large room and kitchenette (sleep 4-5). Some with cooking facilities, room service, restaurants.

Tariff: \$20-\$25 double
\$18 single
\$35 self-contained 2 bedrooms

6. Broadway Motel, 128 Frank Street, (Gold Coast

Highway), Labrador, 4125. (3KMs from conference) Phone 075-31-3288.

Air conditioned, Heat, TV, Fridge, some with cooking facilities, clean and comfortable.

Tariff: \$20-\$30 double

CARAVAN PARKS:

1. HiWay Caravan Park- 6 Frank Street, Southport. 4215. (2KMs from conference) Phone 075-31-2281.

Tariff: \$16 per night (1 or 2 people)
\$5 per extra person

2. Treasure Island Caravan Park- Brisbane Road (Gold Coast Highway), Labrador, 4215. Near the drive-in (5KMs from conference). Phone 075-37-1649.

Tariff: \$23 Double
+\$3 each extra person
* All linen available.

On site vans: \$17 double per day.

+\$2 each extra person
* No linen, but can be hired.
\$20 double (with ensuite)
+\$2 each extra person

3. CoCoNut Grove Holiday Village, (will the CoCoNuts from Victoria stay here?) 2342 Gold Coast Highway, Miami Beach (about 10KMs from conference) Phone 075-55-2577.

Tariff:

On-site vans: \$16 per night (1 or 2 people)
+\$2 each extra person
* No linen or blankets provided;
but can be hired (\$1.75)

Self-contained: \$20 double
+\$2 each extra person
* Blankets provided, but no linen.

4. Palm Beach Caravan Park, 1336 Gold Coast Highway, Palm Beach. (about 13KMs from conference) Phone 075-35.3359.

On site Vans: \$16 double
+\$3 each extra adult;
+\$1.75 each extra child.
* Blankets and linen supplied

TRANSPORT:

Transport is available (private or public) within a short distance of each of the above, along the Gold Coast Highway, if needed.

QUESTIONS:

If you have any questions, please write to me or ring on 075-51-0015. See you all at CoCoConf.

Tutorials.

The following tutorials have been organised, and will be taken by the speaker shown.

1. BASIC Tricks. (Advanced Basic). Tino Delbourgo.
2. 128K & OS9 Bob Thomson.
3. Hardware Modifications to CoCo Brian Dougan

Continued on P 60 ...

EDUCATION PAGE

This month we have programs from America, despite some very useful input from Bob Horne of Ipswich, Qld. We thought that the US programs would compliment the two programs we have available on the tape "Say the Word".

Since when did "Word Pack Speller" become "Say the Word", you may well ask! Well, since we received a program from Oz-Wiz which was too good to leave off! So right from the start, all those who have received the "Say the Word" tape have also received the Oz Wiz program as well.

For those who don't know what I'm talking about, that's alright, I have the same problem. "Say the Word" is a program tape we supply containing two programs which utilise Tandy's Speech - Sound Pack, to assist teachers motivate children with their spelling. The first program has around 500 second grade words, taken directly from the curriculum, which the computer actually speaks. The computer places the word in a sentence, says the word again, and then awaits the child's input. The sentence is displayed on the screen, minus the required word. The input is "lowered" by the computer into the sentence; if particular letters are incorrect, then these are removed by the computer, and the child is asked to spell the word once more. When a correct response is obtained, the computer asks for the next word in the list. No scores are kept.

Words and sentences for other grades can be easily input by the teacher as DATA lines, or you can await our next release, which will probably be for fourth grade.

"Say the Word" is \$39.95 from us or our agents.

Back to Bob Horne.

Bob has produced a series of word games, which have been successfully road tested in his class room. We are highly honoured to have these programs and look forward to featuring the first one next month.

The advent of the Tandy 1000 makes the Educational computer scene very interesting. The Tandy 2000 has been around for a while, and is a finalist in the "Computer of the Year" stakes run by "Your Computer". The 2000 is wonderful for many applications, and I would think that large schools might well consider it as essential.

But the 1000 will make some inroads into the 2000's territory, because it can do many of the same tasks with much of the same software.

We saw OMNUS at the computer show in Sydney, which rammed home this point! OMNUS will control the school files. By that I mean that essentially any information a school is likely to want to record against a pupil, or any report a school is likely to want to receive on students, can be obtained in seconds from this most useful tool.

I'm not going to spend much time on OMNUS at this

point, because we are about to receive our first 1000, and we will review OMNUS in detail when we get the time to look in greater depth at all it can do. Suffice to say, I allowed an hour to see this program put through it's paces at the show, two hours later I had to go and we still hadn't seen all!

If you are impatient for more information, contact Leo Wilson (Tandy, Fortitude Valley) or Karel Davey (Tandy, Mt Druitt).

Our Education section of CoCoLink is now operative, just when visitors are to be excluded from that section. If you are a teacher and wish to access CoCoLink during the day, call me on 075 51 0015 first and I'll arrange for you to get in on a temporary permit! I can only do this in the daytime, when I'm here!

An excellent database called STARS has come to my notice, and I suggest that a subscription to it would be a good investment for a school. It has a heap of info in three main categories - SPELLING HELP, THESAURAS, and SUBJECT SEARCH. We'll talk more about STARS next month. The impatient ones can contact STARS on 02 235 3859.

The prototype CoCoConnection also arrived as we were buttoning up this issue. Have not had a long play with it yet, but it looks well!

Just an initial run with it indicated how easy it was to manipulate external devices by computer with it.

While the magazine is at the printers, we hope to latch CoCoConnection to a model railway - THAT will be the test!

We have begun in earnest our campaign to have CoCo accepted in NSW schools next contract. It was the major reason for my recent trip to Sydney.

It is not going to be easy and we will need support from teachers in that state, so if you want CoCo in your schools, tell me about it!

After NSW, we hope to be in a good position to influence the Education Dept in WA, where the schools have all sorts of computers, but no CoCos!

Finally, if you are a teacher interested in the use of computers in schools, then I would recommend CoCoConf to you. We expect to have several tutorials of interest to you, the first being on "Programming for Children", and the second, a tutorial more philosophically based, on "Computers in the School".

Let CoCo Talk You Into A Better Education

With or without a sound pack, this program can be a useful learning tool.

By Fred B. Scerbo

After several months of fun and graphics, I thought it might be a good idea to get back into some practical programs for your Color Computer. I have been receiving some very practical suggestions lately which will take a little time to develop.

Recently, some readers have suggested I develop something to go along with the speech synthesizer program packs which are now available from a number of distributors, including Radio Shack. So, this month's "Wishing Well" offers an educational program which will put these voice programs to full use, while at the same time fulfilling a wish I received almost two years ago.

The result, *Talking Micro Math Lab*, with or without speech will prove to be a valuable tool to anyone who is interested in teaching youngsters fundamental addition and subtraction skills. Notice I said with or without speech. There are a number of reasons why I designed this program to be used without a sound pack, as well as with.

First, even though the voice packs which are now available are all of excellent quality, only a small percentage of the many CoCo users have actually purchased one. Therefore, I did not want to list a program which could only be used by a small percentage of people.

Secondly, until recently, I have not been an advocate of using "talking computers." I have always thought of them as something of a gimmick. Besides, as a teacher, I had visions of a classroom teacher competing with the voice of a computer for the attention of the students. Little did I dream how useful they can be for educational purposes, especially when they are used with a set of headphones. (No competition, either.)

Therefore, *Talking Micro Math Lab* is designed with three options: 1) a non-talking version, 2) using the new Radio Shack Speech-Sound Pak, and 3) using Spectrum Voice Pak, available from Spectrum Projects. (It might also work

with some other brand voice packs, but at the time of this writing, I did not have access to those other brands. If their distributors can loan me their models, I may be able to make future talking programs in the "Wishing Well" compatible with most other versions, too!) Also, I will include instructions on how to adapt the non-talking version to the MC-10 with the memory expansion pack.

The following listing is designed to work in its non-talking version in 16K Color BASIC, in its Radio Shack Speech-Sound Pak version in 16K, or in its Spectrum version in 32K from tape or disk. I have tried to shrink the Spectrum version down to work in 16K, but it looks like the program is just a little too big to work with the machine language driver it requires. Hopefully, those of you with the Spectrum Voice Pak will have at least 32K.

Now, let's spend a little time looking into the background which led to this month's article.

The Wish

The wish which led to this program actually predates my interest in computer speech. In fact, granting the wish actually came more along the lines of performing a minor miracle. You see, the original *Micro Math Lab* was developed with a specific student of mine in mind. This student's name is Mark, and his story is very interesting, indeed.

Mark and his twin brother, Bobby, came to me as students over two years ago at the age of 17. They had been born with a serious birth defect which left them with somewhat limited abilities. In spite of these serious limitations, both have been an absolute joy to work with. They are the kind of students one gets a really good feeling about being able to help.

I knew what I was getting before Mark and Bobby arrived rather belatedly at the high school level. I had

known both from the time they were born and knew how great a task their parents must have had in trying to work with teachers to educate both in the simplest of life skills. Just before receiving both boys, the family helped fill our high school staff in on what limited skills both had. I was told Bobby was actually quite good in math, while Mark could not even comprehend adding two plus two.

What I later found was that what one was strong in, the other was usually weak in. For example, Bobby could handle rather complicated multiplication and division, but was extremely sloppy in his writing or in something as simple as coloring a picture. Mark, on the other hand, could not add, but was extremely neat and precise at a task such as coloring or handwriting.

Fortunately, what the family did not tell me was their teachers of the last eight years had assured them Mark would *never* learn to add! In fact, they told them to give up!

After working with Mark for just a few days, it became painfully obvious how limited his math skills were. He had developed the habit of using a scrap paper to draw lines or checkmarks to count in adding a single digit addition problem. This made his work slow, with little recall when not using the lines. Ask him what '7' and '9' are, and he would not know!

Therefore, I started working on a colorful math program for the CoCo that would catch Mark's attention as well as slowly draw him away from using lines. If I could have the computer screen display colored blocks corresponding to the digits in a given problem, I could get him to count the blocks on the screen, and then relate the total to the visual image of the math problem.

The result of this effort was the original version of *Micro Math Lab*, which I put Mark to work on for two class periods a day. On the average,

Mark could complete nearly one hundred problems in a 45-minute period. I'll get into how to use the program a little later.

What Were The Results?

Recently, I spoke with Mark's parents. I asked them if they had checked how well Mark was doing in his addition. Since we had just come back from summer vacation, I was not at all surprised they had not given him any busy work over the summer. I promised to send home some math sheets so they could watch Mark doing his math.

The problems I sent home were four-digit, two-number addition problems. To their astonishment, Mark was able to complete the sheet in just minutes with no errors and with no number lines! They called to ask how I did it. It was at this time they told me about the prediction from his previous teachers that Mark would never learn to add.

Since that time we have also worked on subtraction and money handling. Mark's parents are also ready to buy the boys their own CoCo to use at home, and as you can guess, some "Wishing Well" software will probably go along with it.

The program more than proved its worth. However, I have two dozen more students who are in the same condition as Mark was. (Remember, these are handicapped, special needs students.) They have started this comprehensive program using the *Micro Math Lab* to strengthen their addition and subtraction skills.

Since our department just purchased the Radio Shack Speech-Sound Pak, I decided to adapt the program to use the speech it could generate. As you can imagine, some of the students do get a little bored during a long stretch with the computer. The speech would help liven things up.

The Program

The final result is the program you see listed here. As I mentioned, I have made it usable with both Radio Shack's Speech-Sound Pak and the Spectrum Voice Pak. Let me take a moment to mention the difference between the two.

The voice produced by the Radio Shack Pak is a very clear, precise voice without any real inflection, which is not bad for the type of students I am working with. I find the voice quite pleasant, in fact. The Pak does not require a machine language program. It requires a few PEEKs and POKEs, which only make up a few lines in your program.

The Radio Shack program does have two drawbacks. First, it does not read numbers correctly. The number 1,278 would be pronounced "one two seven eight" rather than one-thousand two-hundred seventy-eight. This means strings must be saved to pronounce these numbers, which can be an obstruction in any talking math program. This program takes this into consideration and compensates for it.

Secondly, there is a hefty price of about \$100, almost \$30 to \$40 more than others. If you don't like to have to load in machine language drivers, the extra price may be worth it.

I have found one other drawback, but this may be a quirk in my program pack or in my older CoCos. Every so often, the computer will only produce static instead of speech. This can be resolved by pressing the Reset button. As I said, this may only be something with my own copy, and it is not a serious problem.

Spectrum's version does require a machine language driver, but this driver lets you do much more than the Radio Shack version, such as save a glossary of frequently used words. It also pronounces long numbers in their correct verbal form. Add to this the lower price tag and you have speech at a reasonable price. The voice is a little more "stuffed nosed" than Radio Shack's, but some of my students like it better because it gives the computer a little personality.

Maybe this will help some of you who haven't gotten a voice pack decide which one to get. I like having both. With a little luck, I may get to report to you on the other brands later.

In writing the programs, I wrote strings identified as AS to generate the speech. You will notice I used phonetic spellings of the words I want spoken to get exactly the type of speech I wanted.

This was especially necessary with numbers such as 10 through 19, which the Radio Shack version could not pronounce in their numeric form. These strings will work equally well with the Spectrum version, even though they are not necessary.

If you use the non-talking version, you will not have to worry about the program being totally silent. I have included SOUND commands that are ignored in the talking mode. I have found that SOUND commands disconnect the Radio Shack's Speech Pak for some reason. Therefore, a number of IF/THEN statements will prevent this

from happening.

I have not included the machine language driver for the Spectrum version since I would assume if you have one, you must have the program. You can also load from either tape or disk, if you have a Multi-Pak Interface or Y-cable. Those who have the Spectrum Pak will be familiar with the loading techniques. Once the driver is loaded, it does not have to be reloaded for additional runs.

Using The Program

The first screen that appears will give you the three choices: A) non-talking, B) Radio Shack and C) Spectrum. Choose 'A,' and the program will proceed without sound. Press 'B,' and the proper values for the POKE will be used. Press 'C,' and the program will prompt you for tape or disk. Load the driver and the title card will appear.

If you use the non-talking mode, then the title TALKING will not appear. When in the talking mode, the title will be spoken. Press any key to go to your selections. You may next press 'S' for single digits or 'D' for double digits. Next, press 'A' for addition or 'S' for subtraction.

When your problems appear, you will notice the colored blocks to the right or left of each digit correspond by color and number to the digits in the problems. Therefore, the student can count the blocks to get an answer.

An arrow will point to the column being used. If an answer is 12, then the 2 must be entered first, and then the arrow will move to the next column. If any carrying is required, it will appear as a small carry number at the top of the next column. A green block will also be added for carrying purposes. If an error is made, then the program will make you continue until you get it right.

In the speaking versions, all responses and questions are actually spoken such as "How much is" or "No! The answer is not . . ." When you wish to see the number of correct answers and number of misses, press the '@' key, and this information will appear. Pressing ENTER will *reRUN* the entire program.

MC-10 Version

Unfortunately, there is no way to get a talking version for the MC-10 that I know of. Still, you may be able to get a very nice non-talking version. To correct all the screen poke locations, I have included the variable MC which equals zero in the CoCo version. Adding this line:

```
15 MC=15360
```

will correct the values. You may also wish to delete the following groups of lines:

50 through 150
190 through 210
230
400
500
580
2000 through 5010

You may also wish to neglect any strings known as AS when they appear in a line, but it might be safer just to follow the directions above rather than tampering with the contents of the lines.

Conclusion

Here you have it: a talking educational program I dare say is equal to if not superior to some you may pay for. For my own purposes, I have finally broken into the world of synthetic speech. I

have come up with a few more gems which I may share with you if enough of you are interested, such as a talking multiple choice quiz similar to the screen quiz programs from quite a few months ago.

Best of all, I have been able to achieve something with Mark that others were convinced was impossible. To me, that is the best little miracle of all.

200139	98066
32062	1160236
410194	136022
570165	END195
740220		

The listing:

```

1  '*****
2  '*  TALKING MICRO MATH LAB  *
3  '*    BY FRED B. SCERBO    *
4  '*  COPYRIGHT (C) 1984    *
5  '* 149 BARBOUR ST.N. ADAMS, MA *
6  '*****
10 CLEAR80
20 T=80:K=1:WJ$="PLUSS"
30 DIM Z$(19)
40 CLS
50 PRINT@132,"A) NON-TALKING VER
SION"
60 PRINT@196,"B) SPEECH-SOUND PA
K"
70 PRINT@260,"C) SPECTRUM VOICE
PAK"
80 X$=INKEY$:IF X$="" THEN 80
90 IF X$="A" THEN VP=0:GOTO220
100 IF X$="B" THEN VP=1:GOTO130
110 IF X$="C" THEN VP=2:GOTO3000
120 GOTO80
130 XX=&HFF00:YY=&HFF7E
140 POKEXX+1,52:POKEXX+3,63
150 POKEXX+35,60
160 GOTO220
170 REM CREATE VOICE
180 IF VP=0 THEN RETURN
190 IF VP=1 THEN 2000
200 IF VP=2 THEN 4000
210 RETURN
220 REM START
230 Z$(10)="TEN":Z$(11)="ELEVAN
":Z$(12)="TWELV":Z$(13)="THIRTEE
N":Z$(14)="4 TEEN":Z$(15)="FIFTE
EN":Z$(16)="6 TEEN":Z$(17)="7 TE
EN":Z$(18)="8 TEEN":Z$(19)="9 TE
EN"
240 R$=CHR$(128):Z$=CHR$(207):V$
=CHR$(204)
250 T$=CHR$(197):U$=R$+R$+R$+R$+
R$:UL$=R$+R$+R$:UV$=CHR$(159):UF
$=CHR$(175):UG$=CHR$(255)+CHR$(2

```

55)

```

260 YU$=CHR$(172):GOTO420
270 CLS0:PRINT@0,CHR$(255):;FOR I
=1TO30:PRINT@I,CHR$(252):;NEXT I:
PRINTUG$:;PRINT@63,UG$:;
280 PRINT@70,T$CHR$(203)R$UV$CHR
$(202)CHR$(196)Z$CHR$(200)R$Z$V$
V$R$Z$V$CHR$(203)R$Z$V$Z$U$UG$UL
$R$R$:;
290 PRINTT$CHR$(202)Z$T$CHR$(202
)R$Z$R$R$Z$R$R$R$Z$V$CHR$(203)R$
Z$R$Z$U$UG$UL$R$R$CHR$(196):;
300 PRINTCHR$(200)R$CHR$(196)CHR
$(200)CHR$(196)V$CHR$(200)R$V$V$
V$R$V$R$V$R$V$V$V$U$UG$:;
310 PRINT@167,CHR$(165)CHR$(171)
R$CHR$(167)CHR$(170)R$CHR$(167)Y
U$CHR$(171)R$YU$UF$YU$R$UF$R$UF$
:;
320 PRINT@191,UG$:;
330 PRINT@199,CHR$(165)CHR$(170)
UF$CHR$(165)CHR$(170)R$UF$YU$UF$
R$R$UF$R$R$UF$YU$UF$:;PRINT@223,
UG$:;
340 PRINT@231,CHR$(164)CHR$(168)
R$CHR$(164)CHR$(168)R$YU$R$YU$R$
R$YU$R$R$YU$R$YU$:;PRINT@255,UG$
:;
350 PRINT@266,CHR$(191)R$R$R$CHR
$(183)CHR$(188)CHR$(187)R$CHR$(1
91)CHR$(188)CHR$(187):;PRINT@287
,UG$:;
360 PRINT@298,CHR$(191)R$R$R$CHR
$(191)CHR$(188)CHR$(191)R$CHR$(1
91)CHR$(188)CHR$(187):;PRINT@319
,UG$:;
370 PRINT@330,CHR$(188)CHR$(188)
CHR$(188)R$CHR$(188)R$CHR$(188)R
$CHR$(188)CHR$(188)CHR$(184):;
380 PRINT@351,UG$:;FOR I=1TO30:PR
INTCHR$(243):;NEXT:PRINTCHR$(255
):;
390 IF VP=0 THEN RETURN
400 PRINT@66,CHR$(245)CHR$(252)C
HR$(250):;WW$="TALKING":FORWW=1T
O7:EW=ASC(MID$(WW$,WW,1)):PRINT@
WW*32+66,CHR$(245)CHR$(EW+32)CHR
$(250):;NEXTWW:PRINT@WW*32+66,C
HR$(245)CHR$(243)CHR$(250):;RETU
RN

```

```

410 A$="MY CRO.MATH.LAB":GOSUB17
0:RETURN
420 GOSUB270:GOSUB410:FORA=0TO9:
READ E,F,G:A$(A)=CHR$(E)+CHR$(F)
+CHR$(G):D$(A)=CHR$(E-T)+CHR$(F-
T)+CHR$(G-T)
430 READ E,F,G:B$(A)=CHR$(E)+CHR
$(F)+CHR$(G):E$(A)=CHR$(E-T)+CHR
$(F-T)+CHR$(G-T)
440 READ E,F,G:C$(A)=CHR$(E)+CHR
$(F)+CHR$(G):F$(A)=CHR$(E-T)+CHR
$(F-T)+CHR$(G-T):NEXTA
450 PRINT@417,"by"+R$+"fred"+R$+
"scerbo"+R$+R$+"copyright";:POKE
1467+MC,49:POKE1468+MC,57:POKE14
69+MC,56:POKE1470+MC,52
460 IFINKEY$=""THEN460
470 GOSUB270
480 PRINT@420,"s"+R$+"ingle"+R$+
"or"+R$+R$+"d"+R$+"ouble"+R$+"di
gits";
490 POKE1443+MC,40:POKE1445+MC,4
1:POKE1451+MC,32:POKE1454+MC,32:
POKE1455+MC,40:POKE1457+MC,41:PO
KE1463+MC,32
500 A$="PRESS S FOR SINGUL DIJJI
TS OR D FOR DUBBUL DIJJITS":GOSU
B170
510 X$=INKEY$:IFX$=""THEN510
520 IFX$="S"THEN560
530 IFX$="D"THEN550
540 GOTO510
550 D=2
560 PRINT@452,"a"+R$+"ddition"+R
$+"or"+R$+R$+"s"+R$+"ubtraction"
;
570 POKE1475+MC,40:POKE1477+MC,4
1:POKE1485+MC,32:POKE1488+MC,32:
POKE1489+MC,40:POKE1491+MC,41
580 A$="PRESS A FOR ADDITION OR
S FOR SUBTRACTION":GOSUB170
590 X$=INKEY$:IFX$=""THEN590
600 IFX$="A"THEN820
610 IFX$="S"THEN630
620 GOTO590
630 K=-1:WJ$="MY NUS":GOTO820
640 PRINT@P,A$(R);:PRINT@P+32,B$
(R);:PRINT@P+64,C$(R);:RETURN
650 PRINT@P,D$(R);:PRINT@P+32,E$
(R);:PRINT@P+64,F$(R);:RETURN
660 PRINT@AP,CHR$(209)CHR$(219);
:PRINT@AP+31,CHR$(209)CHR$(216)C
HR$(218)CHR$(217);
670 PRINT@AP+65,CHR$(218);:RETUR
N
680 PRINT@AP,R$R$R$;:PRINT@AP+31
,R$R$R$R$;:PRINT@AP+64,R$R$R$;:R
ETURN
690 FL=FL+1:A$="NO.THEE ANSWER I
S NOT "+X$:GOSUB170:FORHH=1TO600

```

```

:NEXTHH
700 IF VP=0 THEN SOUND40,2
710 RETURN
720 X$=INKEY$:IFX$=""THEN720
730 IFX$="@"THEN1250
740 X=ASC(X$)-48:IFX<0THEN720
750 IFX>9THEN720
760 RETURN
770 IF TB+(BB*K)>9 THEN X$=Z$(TB
+BB*K)+"CAIRY ONE"
780 IF X=TA+(BA*K)-(K*10) THEN X
$=Z$(X+10)
790 A$="CORRECT.THEE ANSWER IS "
+X$:GOSUB170
800 IF VP=0 THEN SOUND200,1:SOUN
D200,1
810 RETURN
820 CLS0:FORHH=1TO1600:NEXTHH:TA
=0:AT=AT+1
830 IFD=0THEN850
840 TA=RND(9):BA=RND(9):IFBA=>TA
THEN840
850 TB=RND(10)-1:BB=RND(10)-1:IF
BB=>TB THEN850
860 IFD=0THEN880
870 R=TA:P=44:GOSUB640:R=BA:P=14
0:GOSUB650
880 R=TB:P=48:GOSUB640:R=BB:P=14
4:GOSUB650
890 PRINT@231,"";:FORI=1TO16:PRI
NTCHR$(156);:NEXTI
900 IFK=-1THEN930
910 PRINT@168,CHR$(151)CHR$(155)
;:PRINT@200,CHR$(148)CHR$(152);
920 GOTO940
930 PRINT@168,CHR$(147)CHR$(147)
CHR$(146);
940 AP=400:GOSUB660
950 IFD=0THEN1000
960 IFTA=0THEN980
970 FORI=1TOTA:PRINT@1+I*32,CHR$
(252);:NEXT
980 IFBA=0THEN1000
990 FORI=1TOBA:PRINT@3+I*32,CHR$
(172);:NEXT
1000 IFTB=0THEN1020
1010 FORI=1TOTB:PRINT@30+I*32,CH
R$(252);:NEXT
1020 IFBB=0THEN1040
1030 FORI=1TOBB:PRINT@28+I*32,CH
R$(172);:NEXT
1040 A$="HOW MUCH IS "+STR$(TB)+
WJ$+STR$(BB):GOSUB170:GOSUB720
1050 P=272:R=X:GOSUB640
1060 IFX=TB+(BB*K)THEN1090
1070 IFX=TB+(K*BB)-(K*10)THEN109
0
1080 AP=P:GOSUB690:GOSUB680:GOTO
1040
1090 GOSUB770:AP=400:GOSUB680:AP

```

```

=396:GOSUB660
1100 IF TB+(BB*K)>9 THEN TB=0:BB
=0:POKE1036+MC,49:PRINT@1+32*TA,
CHR$(140):A$="HOW MUCH IS ONE P
LUSS"+STR$(TA)+WJ$+STR$(BA):GOSU
B170:GOSUB720:TA=TA+1:GOTO1130
1110 IFTA=0THEN GOSUB680:GOTO124
0
1120 A$="HOW MUCH IS "+STR$(TA)+
WJ$+STR$(BA):GOSUB170:GOSUB720
1130 P=268:R=X:GOSUB640
1140 IFX=TA+(BA*K)THEN1170
1150 IFX=TA+(BA*K)-(K*10)THEN118
0
1160 AP=P:GOSUB690:GOSUB680:GOTO
1110
1170 GOSUB770:FORI=1TO2000:NEXT:
GOTO820
1180 GOSUB770:AP=396:GOSUB680:AP
=392:GOSUB660
1190 GOSUB720
1200 P=264:R=X:GOSUB640
1210 IFX=1THEN1230
1220 AP=P:GOSUB690:GOSUB680:GOTO
1190
1230 GOSUB770
1240 FORI=1TO2000:NEXT:GOTO820
1250 CLS:GOSUB270:PRINT@419,"co
rrect "R$;AT-1;R$R$"misses "R$;FL;
1260 IFINKEY$=CHR$(13) THEN RUN
1270 GOTO1260
1280 DATA 247,252,251,255,240,25
5,244,252,248
1290 DATA 241,255,240,240,255,24
0,244,252,248
1300 DATA 254,252,251,243,252,24
1,252,252,252
1310 DATA 252,252,251,252,252,25
5,252,252,248
1320 DATA 255,245,250,252,253,25
4,240,244,248

```

```

1330 DATA 255,252,252,252,252,25
5,252,252,252
1340 DATA 255,252,252,255,252,25
5,252,252,252
1350 DATA 254,252,255,240,247,24
8,244,248,240
1360 DATA 255,252,255,255,252,25
5,252,252,252
1370 DATA 255,252,255,252,252,25
5,252,252,252
2000 FORI=1TOLEN(A$)
2010 IF PEEK(Y)AND 128=0 THEN20
10
2020 POKEYY,ASC(MID$(A$,I,1))
2030 NEXTI
2040 IFPEEK(Y)AND128=0THEN2040
2050 POKEYY,13
2060 FORHH=1TO900:NEXTHH:RETURN
3000 IF PEEK(&H6000)=&H7F AND PE
EK(&H6005)=&H20 THEN 220
3010 GOTO5000
3020 CLEAR1000,&H5FFF
3030 PRINT@327,"(D)ISK OR (T)APE
"
3040 X$=INKEY$:IFX$="D"THEN3050E
LSEIFX$="T"THEN3060ELSE3040
3050 CLS:PRINT@233,"PLEASE STAND
BY";:LOADM"TRNSLATE":POKE&HFF40
,0:GOTO3070
3060 CLS:PRINT@233,"PLEASE STAND
BY";:CLOADM"TRNSLATE"
3070 DEFUSR1=&H6000:DEFUSR2=&H60
05:DEFUSR3=&H6007
3080 T=80:K=1:WJ$="LUSS":VP=2:D
IM Z$(19)
3090 GOTO220
4000 X$=USR2(A$)
4010 FORHH=1TO700:NEXTHH:RETURN
5000 PCLEAR1
5010 GOTO3020

```

**BEST
OF**

COCOOLZ

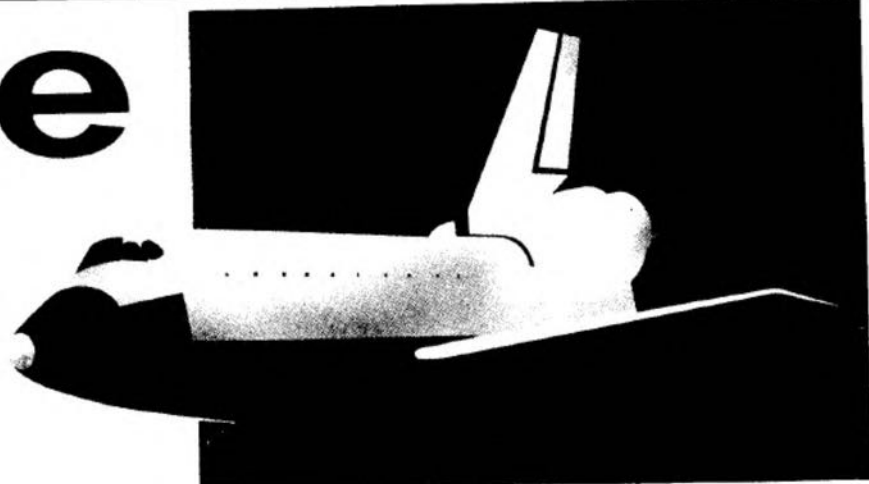
#1

★ A selection of
14 wide-ranging
EDUCATIONAL programs
for the CoCo

★ GREAT FOR THE FAMILY LIBRARY

tape **\$10** or disk **\$21.95**

Space Race



By Daniel Hamilton

Welcome to *Space Race*. This is an interesting math game with a new twist. Instead of answering a math problem, you must enter an equation and the CoCo computes the answer. I borrowed the technique to accomplish this from Rich Dersheimer's *Mathgame* program published in the January 1984 RAINBOW.

The object of the game is simple: be the first player to pilot your spaceship from Earth to Pluto. Each player's turn consists of spinning for three random numbers, then building an equation from the numbers and the arithmetic operators: * / + -. You may only use each number or operator once to form your equation. When your equation is entered, the CoCo computes its value, and moves your spaceship a distance equal to this value. Only the integer portion of the value is used, and values less than zero are ignored. There are two rules that make the game more interesting.

- 1) If you stop on a planet, you automatically advance to the next planet.
- 2) If you stop on the same position as your opponent, your opponent is sent back to the previous planet.

This means the equation with the highest value is not necessarily the one that will most improve your position in the race. An exception to rule #2 is that planets are considered safety areas; you cannot be bumped back while you are on a planet.

The game can be played by two players, by one person against the computer, or enter "Computer" as the name of the first player and select the one player option, and the computer

will play a demonstration game against itself.

I must apologize for the lack of remarks and all the multiple statement lines. I abhor multiple statement lines because they make the logic difficult to follow, especially in lines with multiple *IF* and *ELSE*.

Unfortunately, the program as I originally wrote it ran over 16K, and I really wanted a game everyone could try, so I packed it all together and stripped out all the remarks to get a version that just barely runs in 16K.

A breakdown of the program follows:

Line 1 Dimension arrays: reseed random function
Line 2 Dummy line to compute input equation
Line 3-4 Locates dummy line
Line 5-9 Initialize variables: input names: select mode of play
Line 10 Draws screen
Line 11 Alternates players (beginning of main loop)
Line 12 Spins for 3 random numbers
Line 13 Stores numbers and operators and draws them on screen

Line 14-17 Gets input equation
Line 18-20 Checks syntax of input equation
Line 21 Inserts input equation in dummy equation
Line 22-24 Evaluate equation and move rocket (end of main loop)
Line 25-26 Subroutine to move player 1 rocket
Line 27-28 Subroutine to move player 2 rocket
Line 29-30 Subroutine to bump back player 2
Line 31-32 Subroutine to bump back player 1
Line 33-38 Subroutine to spin for 3 random numbers
Line 39-44 Subroutine to build computer's equation
Line 45-47 Subroutine to draw screen
Line 48 Subroutine to draw text on screen
Line 49-54 Defines text character strings: defines music strings
Line 55-63 Draws rockets and stores them in arrays
Line 64-65 Data
Line 66-67 'End of game' routine

108	42190
2217	50110
27103	53154
35107	END96

The listing:

```
1 CLEAR150: DIMC$(58), A(1,6), B(1,6), C(1,6), D(1,6), E(1,6), F(1,6), G(1,6), H(1,6), M$(2), N$(2), P(2), Z(7), N(3): N=RND(-TIMER): GOTO3
2 V=*****: RETURN
```

```

3 E=PEEK(25)*256+PEEK(26)
4 IFPEEK(E)=173THEN5ELSESEE=E+1:GO
  TO4
5 CLS:PRINT@267,"space"+CHR$(12
  8)+"race";:SCREEN0,1:GOSUB49
6 A=32:B=8:C=32:D=15:P(1)=0:P(2)
  =0:F=0:CLS:PRINT@256,"";:INPUT"E
  NTER YOUR NAME";N$(1):IFN$(1)=""
  THENN$(1)="PLAYER 1"ELSEIFLEN(N$
  (1))>8THENN$(1)=LEFT$(N$(1),8)
7 CLS:PRINT@256,"PRESS: 1) TO R
  ACE THE COMPUTER";:PRINT@328,"2)
  TO RACE A FRIEND";
8 A$=INKEY$:IFA$<"1"ORA$>"2"THEN
  ELSEIFA$="1"THENN$(2)="COMPUTER
  ":GOTO10
9 CLS:PRINT@256,"";:INPUT"ENTER
  YOUR FRIEND'S NAME";N$(2):IFN$(2)
  )=""THENN$(2)="PLAYER 2"ELSEIFLE
  N(N$(2))>8THENN$(2)=LEFT$(N$(2),
  8)
10 GOSUB45
11 IFF<>1THENF=1ELSEF=2
12 GOSUB33
13 FORI=1TO3:Z(I)=N(I)+48:NEXTI:
  Z(4)=42:Z(5)=47:Z(6)=43:Z(7)=45:
  LINE(40,152)-(255,191),PRESET,BF
  :DRAW"BM48,160":FORI=1TO7:DRAWC$
  (Z(I)-32)+C$(0):NEXTI:IFN$(F)="C
  OMPUTER"THENGOSUB39:GOTO21
14 G=40:H=183:T$="ENTER YOUR EQU
  ATION":GOSUB48:H=191:T$="PRESS C
  LEAR TO MAKE CHANGES":GOSUB48:L=
  48:T=0:E$=""
15 B$="BM"+STR$(L)+",173;"
16 A$=INKEY$:DRAWB$+"NR0C0NR0C1"
  :IFA$=""THEN16ELSEIFT=5THEN17ELS
  EFORI=1TO7:IFASC(A$)=Z(I)THENDRA
  W"BM"+STR$(32+I*16)+",160"+C$(0)
  :DRAWB$+C$(Z(I)-32):E$=E$+A$:Z(I)
  )=0:L=L+8:T=T+1ELSENEXTI
17 IFA$=CHR$(12)THEN13ELSEIFA$=C
  HR$(13)ANDT=5THEN18ELSE15
18 DRAWC$(0)+C$(29):FORI=1TO5STE
  P2:IFMID$(E$,I,1)<"0"THEN19ELSEN
  EXTI:FORI=2TO4STEP2:IFMID$(E$,I,
  1)>"0"THEN19ELSENEXTI:GOTO21
19 LINE(40,175)-(255,191),PRESET
  ,BF:G=40:H=183:T$="THIS EQUATION
  HAS AN ERROR":GOSUB48:H=191:T$=
  "PRESS CLEAR TO START OVER":GOSU
  B48
20 A$=INKEY$:IFA$=""THEN20ELSEIF
  A$<>CHR$(12)THEN19ELSE13
21 FORI=0TO4:A$=MID$(E$,I+1,1):I
  FA$>"0"THENPOKEE+I,ASC(A$)ELSEIF
  A$="+"THENPOKEE+I,171ELSEIFA$="-"
  THENPOKEE+I,172ELSEIFA$="*"THEN
  POKEE+I,173ELSEIFA$="/"THENPOKEE
  +I,174

```

```

22 NEXTI:GOSUB2:V$=STR$(V):FORI=
  1TOLEN(V$):DRAWC$(ASC(MID$(V$,I,
  1))-32):NEXTI:FORI=0TO4:POKEE+I,
  173:NEXTI:LINE(40,175)-(255,191)
  ,PRESET,BF:IFV<1THEN11ELSEFORI=P
  (F)+1TOP(F)+INT(V):ONF GOSUB25,2
  7:IFI=300THEN66ELSENEXTI:P(F)=P(
  F)+INT(V)
23 IFF(F)/50=INT(P(F)/50)THENG=4
  8:H=183:T$="** BONUS **":GOSUB48
  :PLAYM$(0):FORI=1TO50:ONF GOSUB2
  5,27:NEXTI:P(F)=P(F)+50:IFF(F)=3
  00THEN66
24 IFF(1)=P(2)THENIFF(1)/50=INT(
  P(1)/50)THEN11ELSEONF GOSUB29,31
  :GOTO11ELSE11
25 FORY=32TO128STEP48:IFY=B THEN
  26ELSENEXTY:FORJ=1TO2:PUT(A,B)-(
  A+15,B+6),C,PSET:PUT(A,B)-(A+15,
  B+6),A,PSET:A=A+2:NEXTJ:IFA=232T
  HENLINE(A,B)-(A+15,B+6),PRESET,B
  F:B=B+24:A=A-20:PUT(A,B)-(A+15,B
  +6),E,PSET:RETURNELSERETURN
26 FORJ=1TO2:PUT(A,B)-(A+15,B+6)
  ,G,PSET:PUT(A,B)-(A+15,B+6),E,PS
  ET:A=A-2:NEXTJ:IFA=12THENLINE(A,
  B)-(A+15,B+6),PRESET,BF:B=B+24:A
  =A+20:PUT(A,B)-(A+15,B+6),A,PSET
  :RETURNELSERETURN
27 FORY=39TO135STEP48:IFY=D THEN
  28ELSENEXTY:FORJ=1TO2:PUT(C,D)-(
  C+15,D+6),D,PSET:PUT(C,D)-(C+15,
  D+6),B,PSET:C=C+2:NEXTJ:IFC=232T
  HENLINE(C,D)-(C+15,D+6),PRESET,B
  F:D=D+24:C=C-20:PUT(C,D)-(C+15,D
  +6),F,PSET:RETURNELSERETURN
28 FORJ=1TO2:PUT(C,D)-(C+15,D+6)
  ,H,PSET:PUT(C,D)-(C+15,D+6),F,PS
  ET:C=C-2:NEXTJ:IFC=12THENLINE(C,
  D)-(C+15,D+6),PRESET,BF:D=D+24:C
  =C+20:PUT(C,D)-(C+15,D+6),B,PSET
  :RETURNELSERETURN
29 LINE(C,D)-(C+15,D+6),PRESET,B
  F:FORY=39TO135STEP48:IFY=D THENP
  (2)=P(2)-(212-C)/4:C=212:PUT(C,D)
  -(C+15,D+6),F,PSETELSENEXTY:P(2)
  )=P(2)-(C-32)/4:C=32:PUT(C,D)-(C
  +15,D+6),B,PSET
30 G=48:H=183:PLAYM$(1):T$="TOO
  BAD FOR "+N$(2):GOSUB48:RETURN
31 LINE(A,B)-(A+15,B+6),PRESET,B
  F:FORY=32TO128STEP48:IFY=B THENP
  (1)=P(1)-(212-A)/4:A=212:PUT(A,B)
  -(A+15,B+6),E,PSETELSENEXTY:P(1)
  )=P(1)-(A-32)/4:A=32:PUT(A,B)-(A
  +15,B+6),A,PSET
32 G=48:H=183:PLAYM$(1):T$="TOO
  BAD FOR "+N$(1):GOSUB48:RETURN
33 LINE(40,0)-(255,6),PRESET,BF:
  LINE(40,152)-(255,191),PRESET,BF

```

```

34 FORJ=1TO2:P#=STR$(P(J)):G=40+
(J-1)*112:H=6:T#=N$(J):GOSUB48:D
RAWC$(29):FORI=2TOLEN(P#):DRAWC$
(ASC(MID$(P#,I,1))-32):NEXTI,J:G
=176:H=167:T#=N$(F)+"'S":GOSUB48
:H=175:T#="TURN":GOSUB48:H=183:T
#="TO SPIN.":GOSUB48
35 FORI=60TO140STEP40:CIRCLE(I,1
68),10:NEXTI:IFN$(F)="COMPUTER"
HENG=56:H=191:T#="*SPINNING*":GO
SUB48:GOTO37ELSEG=40:H=191:T#="P
RESS ENTER TO SPIN.":GOSUB48
36 A#=INKEY$:IFA#<>CHR$(13)THEN3
6
37 LINE(40,184)-(255,191),PRESET
,BF
38 FORI=1TO3:FORJ=1TO10:N=3*RND(
3)-3+I:DRAW"BM"+STR$(18+I*40)+",
171;"+C$(0)+"BL8"+C$(16+N):PLAY"
L25501C":NEXTJ:N(I)=N:PLAY"05CBC
":NEXTI:FORI=1TO250:NEXTI:RETURN
39 G=56:H=183:T#="*THINKING - ST
AND BY*":GOSUB48:IFF=1THEN0=2ELS
EO=1
40 R=P(F)-P(O):S=P(O)-(INT(P(O)/
50)*50):M=0:RESTORE
41 FORX=1TO6:READI,J,K:V=N(I)*N(
J)+N(K):P=42:Q=43:GOSUB43:V=N(I)
*N(J)-N(K):Q=45:GOSUB43:V=N(I)*N
(J)/N(K):Q=47:GOSUB43:V=N(I)-N(J)
*N(K):P=45:Q=42:GOSUB43:V=N(I)-
N(J)+N(K):Q=43:GOSUB43:V=N(I)-N(
J)/N(K):Q=47:GOSUB43:V=N(I)/N(J)
+N(K):P=47
42 Q=43:GOSUB43:V=N(I)/N(J)-N(K)
:Q=45:GOSUB43:NEXTX:LINE(40,161)
-(255,183),PRESET,BF:G=48:H=173:
T#=E#+ "=" :GOSUB48:RETURN
43 V=INT(V):IFV<=0THENRETURNELSE
IFP(F)+V=P(O)ANDS>R+M THENM=S-R:
GOTO44ELSEIF(P(F)+V)/50=INT((P(F)
+V)/50)ANDP(F)+V+50>M+P(F) THENM
=V+50:GOTO44ELSEIFV>M THENM=V:GO
TO44ELSERETURN
44 E#=CHR$(N(I)+48)+CHR$(P)+CHR$
(N(J)+48)+CHR$(Q)+CHR$(N(K)+48):
RETURN
45 PMODE3,1:PCLS:FORI=12TO156STE
P24:READX:CIRCLE(X,I),14,,.9:PAI
NT(X,I),RND(2)+1,4:NEXTI:PMODE4,
1:SCREEN1,1:FORH=31TO175STEP24:R
EADG,T#:GOSUB48:NEXTH:FORI=23TO1
19STEP48:LINE(36,I)-(240,I),PSET
:LINE(16,I+24)-(220,I+24),PSET
46 FORJ=0TO49:IFINT(J/5)=J/5THEN
K=2ELSEK=0
47 LINE(240-J*4,I+1+K)-(240-J*4,
I-1),PSET:LINE(16+J*4,I+25+K)-(1
6+J*4,I+23),PSET:NEXTJ,I:PMODE3,
1:COLOR1,4:FORH=14TO158STEP24:RE

```

```

ADG,T#:GOSUB48:NEXTH:COLOR4,1:PM
ODE4,1:PUT(32,0)-(47,14),A,PSET:
PUT(32,15)-(47,21),B,PSET:RETURN
48 DRAW"BM"+STR$(G)+", "+STR$(H)+
";":FORK=1TOLEN(T#):DRAWC$(ASC(M
ID$(T#,K,1))-32):PLAY"L10005E":N
EXTK:PLAY"C":RETURN
49 C$(0)="C0U6RD6RU6RD6RU6RD6BR3
C1":C$(7)="BR2BU4U2RD2BD4BR5":C$
(10)="BR2U6D3NH2NG2NE2F2BDBR4":C
$(11)="BU3R5L3ND2U2RD4BDBR5":C$(
13)="BU3R5BD3BR3":C$(14)="URDBR7
":C$(15)="E5BD5BR3":C$(16)="BUU4
NF4ER3FD4GL3BR7"
50 C$(17)="BR3RU6NGD6RBR3":C$(18
)="BU5ER3FDG2L2GDR5BR3":C$(19)="
BU5ER3FDGNLFDGL3HBD8R8":C$(20)="
BU6D3R4NU3NRD3BR4":C$(21)="BU6NR
5D2R4FD2GL3HBD8R8":C$(22)="BUU4E
R3FBD2BLNL3FDGL3BR7":C$(23)="BU6
R5DG4DBR7":C$(24)="BUUEHUER3FDGN
L2FDGL3BR7"
51 C$(25)="BUFR3EU4HL3GDFR4BD3BR
3":C$(29)="BU2R5BU2L5BD4BR8":C$(
33)="U5ER3FD2NL4D3BR3":C$(34)="U
6R5FDGNL3FDGL4BR8":C$(35)="BUU4E
R3FBD4GL3BR7":C$(36)="U6R3F2D2G2
L3BR8":C$(37)="U6NR5D3NR4D3R5BR3
":C$(38)="U3NR4U3R5BD6BR3"
52 C$(39)="BUU4ER3FBD2NL2D2GL3BR
7":C$(40)="U6BR5D3NL5D3BR3":C$(4
1)="BR2R2LU6LR2BD6BR4":C$(42)="B
U2DFR3EU5BD6BR3":C$(43)="U6BR5G4
EF3BR3":C$(44)="NU6R5BR3":C$(45)
="U6F2RE2D6BR3":C$(46)="U6F5DU6B
D6BR3":C$(47)="BUU4ER3FD4GL3BR7"
:C$(48)="U6R4FDGL4D3BR8"
53 C$(49)="BUU4ER3FD3GNHNFGL2BR7
":C$(50)="U6R4FDGL3RF3BR3":C$(51
)="BUFR3EUHL3HUER3FBD5BR3":C$(52
)="BR2U6L2R5L2D6BR5":C$(53)="BUU
5BR5D5GL3BR7":C$(54)="BU6D2BFD8F
DRUBEUBEU2BD6BR3":C$(55)="NU6E2R
F2NU6BR3":C$(56)="UE4RUBL5DRF4DB
R3"
54 C$(57)="BU6DF2ND3RND3E2UBD6BR
3":C$(58)="BU6R5DG5R5BR3":M$(0)=
"T4L803GP8L32GP32GP3204L4CT2":M$
(1)="T3L401BB-AA-L2GT2":M$(2)="T
1202L4AP4L4AA03L1C02A03L3C02A03C
L1ECL3ECELE1G02GL303C02G03CL1ET2"
55 FORX=1TO6:READI,J,K:NEXTX:FOR
I=1TO8:PMODE3,1:PCLS:READA#:DRAW
A#:PMODE4,1:ONI GOSUB56,57,58,59
,60,61,62,63:NEXTI:RETURN
56 GET(0,0)-(15,6),A:RETURN
57 GET(0,0)-(15,6),B:RETURN
58 GET(0,0)-(15,6),C:RETURN
59 GET(0,0)-(15,6),D:RETURN
60 GET(0,0)-(15,6),E:RETURN

```



```

61 GET(0,0)-(15,6),F:RETURN
62 GET(0,0)-(15,6),G:RETURN
63 GET(0,0)-(15,6),H:RETURN
64 DATA1,2,3,2,3,1,3,1,2,1,3,2,2
,1,3,3,2,1,"BM3,6;C2E3NH3L2R8L2U
L4D2R4","BM3,6;C3E3NH3L2R8L2UL4D
2R4","BM3,6;C2E3NH3C4NL6C2R6L2UL
4D2R4","BM3,6;C3E3NH3C4NL6C3R6L2
UL4D2R4","BM10,6;C2H3NE3R2L8R2UR
4D2L4","BM10,6;C3H3NE3R2L8R2UR4D
2L4"
65 DATA"BM10,6;C2H3NE3C4NR6C2L6R
2UR4D2L4","BM10,6;C3H3NE3C4NR6C3

```

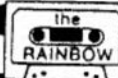
```

L6R2UR4D2L4",16,240,16,240,16,24
0,16,1,EARTH,223,MARS,1,JUPITER,
207,SATURN,1,URANUS,199,NEPTUNE,
1,PLUTO,14,0,234,50,6,100,230,15
0,6,200,230,250,6,300
66 PLAYM$(2):PCLS:G=32:H=100:T$=
"HOORAY HOORAY HOORAY":GOSUB
48:H=116:G=(88-LEN(N$(F))*8)/2:T
$=N$(F)+" WINS THE SPACE RACE":G
OSUB48:G=24:H=148:T$="PRESS ENTE
R TO PLAY AGAIN.":GOSUB48
67 A$=INKEY$:IFA$<>CHR$(13)THEN6
7ELSERUN

```

GAME

16K
ECB



Looie Wants You To Join The Penguin Patrol

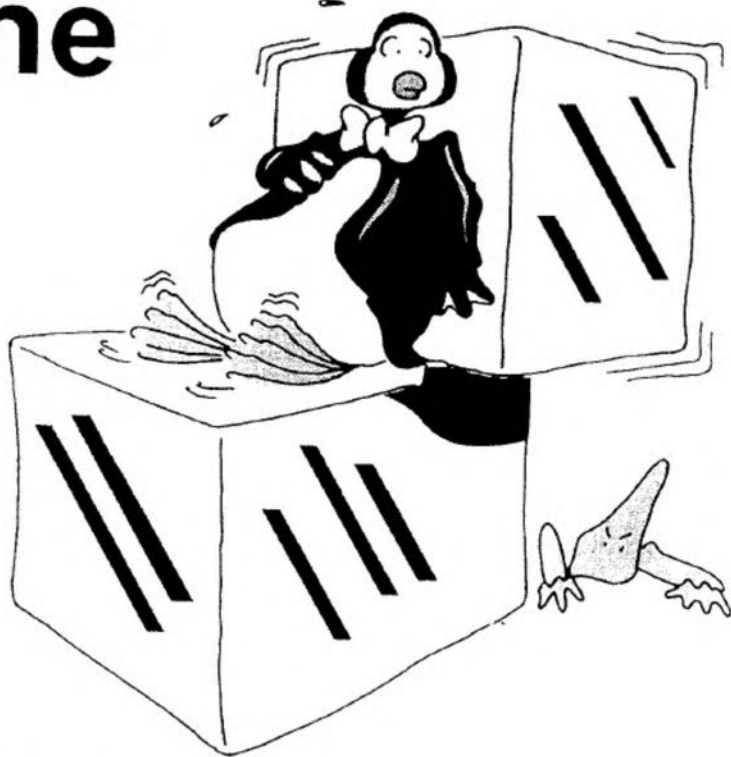
By Paul Wagorn

Those of you who regularly visit your downtown arcade already know what fun it is to put your quarter in your favorite "Pengo" game and try to push the ice blocks into those mean sno-bees. Well, *Penguin* is somewhat like that, except you don't have to put in quarters! It's an action-packed, Hi-Res arcade-style game.

Penguin goes like this: With the four arrow keys you control a penguin named Looie, who is being chased by a pair of coneheads whose only function is to tread him into the ground.

It may seem that everything is pointing towards your destruction, but, there is some hope for your survival. In your little world, you are surrounded by ice blocks which you can push into those meanie coneheads. If one of the ice blocks is in your path, you can simply walk over it. The coneheads, on the other hand, cannot just walk over the ice blocks — they eat them. After

April, 1985



one of the coneheads eats an ice block, he must rest for a short while for fear of getting cramps.

While all of this is going on, a timer is slowly ticking away. You have approximately 20 seconds until a tone starts to beep. After the tone starts to beep, you have another eight seconds to destroy both coneheads, or face the consequences!

You also have something else going for you. If you bump into a wall, any coneheads lined up on that wall are temporarily stunned, giving you time to squash the innards out of them.

AUSTRALIAN RAINBOW

One conehead is better than the other; he moves faster, doesn't have to rest as long after eating an ice block, and stays stunned for less time than the inferior conehead (not the kind of guy you'd want to meet in a dark alley!).

The speedup *POKE* is used in Line 7, so if your computer hangs up with the speedup *POKE*, take it out.

I hope you have as much fun playing *Penguin* as I do. One more thing: good luck, you'll need it!

PAGE 15

12	74
21	215
35	97
51	133
70	239
END	47

The listing:

```

0 POKE 65494,0
1 CLS:PRINT:PRINT:PRINT:PRINT:PRINT
  PENGUIN
  BY
  APOLLO":FORI=5
9TO100:PLAY"T=I;O5;A-;C;D":SCREEN
N0,1:SCREEN0,0:NEXTI
2 PLAY"O3"
3 PLAY"T4;CAGAFEAADFDEGADAP1T4O3
L4AP14L2AL4GFL4CL3CL4DFB-AP4FGO4
L4DCO3AB-AAGL3B-LBAL2F"
4 CLEAR3,16383
5 DIMB(4,4),BG(4,4),BL(3,3),SG(4
,4),DB(3,3)
6 COLOR3,1
7 POKE65495,0:
8 PMODE1,1:PCLS
9 DRAW"SBBM4,14C2FR3URLD2R4L4UH2
F2DR2U4D4R2UE2G2LRDR3EDC1D5C3DGL
10HUE3R6F3":PAINT(18,26),2,3
10 PSET(13,26,3):PSET(15,26,3):P
SET(20,27,3):PSET(22,27,3):PSET(
18,28,4)
11 GET(2,2)-(32,32),SG:PCLS
12 DRAW"SBBM14,4C2R3F3D7GL7HU7E2
G3C3D4GU4GD5C1DBM+13,0U4C3U4FD4F
U4"
13 PAINT(16,8),2,2:PSET(14,10,3)
:PSET(14,12,3):PSET(20,10,3):PSE
T(20,12,3):PSET(16,16,3):PSET(18
,16,3):PSET(12,18,4):PSET(14,20,
4):PSET(16,20,4):PSET(18,20,4):P
SET(20,20,4):PSET(22,18,4):PSET(
8,28,3):PSET(8,30,3):PSET(8,32,3
):PSET(6,32,3)
14 PSET(4,32,3)
15 PSET(24,28,3):PSET(24,30,3):P
SET(24,32,3):PSET(22,32,3):PSET(
26,32,3):PSET(28,32,3):PSET(10,3
2,3)
16 GET(2,2)-(32,32),B
17 PCLS:DRAW"SBC3BM16,8R2F3DFDFD
G2L8H2UEUEUE3LC1L5D11R5C4DL3R5C1
RC4R4L3U":PAINT(18,20),2,3:PSET(
14,18,4):PSET(22,18,4):PSET(12,2
4,3):FORI=14TO22:PSET(I,22,3):NE
XT:PSET(24,24,3)
18 GET(2,2)-(32,32),BG
19 PCLS:COLOR3,1:LINE(8,8)-(26,2
6),PSET,BF:COLOR2,1:LINE(10,10)-
(24,24),PSET,B:COLOR4,1:LINE(12,
12)-(22,22),PSET,BF:COLOR3,1:LIN
E(14,14)-(20,20),PSET,BF:PAINT(1
5,15),4,3:CIRCLE(16,16),3,1,1,0,
.25:GET(2,2)-(28,28),BL
20 PCLS:FORI=1TO47:Y=(RND(6)-1)*
32+4:X=(RND(7)-1)*32+4:PUT(X,Y)-
(X+28,Y+28),BL:NEXT
21 TIMER=0:SCREEN1,0
22 X=(RND(7)-1)*32:Y=(RND(6)-1)*
32
23 X1=(RND(7)-1)*32:Y1=(RND(6)-1
)*32:IF X1=X AND Y1=Y THEN 23
24 X2=(RND(7)-1)*32:Y2=(RND(6)-1
)*32:IF(Y2=Y AND X2=X)OR(X2=X1
AND Y2=Y1) THEN 24
25 LINE(235,0)-(255,191),PSET,B
26 DRAW"BM238,18C3R4FD3GL4U5D10C
1DC3R5L5D4R2L2D4R5L5DC1DC3D7U7F6
DU7D8C1DC3L2GL6D6FR3EU3L3R4C1DBL
2C3D7GL3HU7D8C1D3C3R5L2D7LU7D7L2
R5D1C1D1L5C3D7U7F6DU7"
27 PUT(X1,Y1)-(X1+32,Y1+32),BG:P
UT(X2,Y2)-(X2+32,Y2+32),BG:PUT(X
,Y)-(X+32,Y+32),B:PLAY"T3O1;L4EL
8GB02GL4.EL4F#L8EDP8DP8P8O1L4EL8
GB02GL4.EL4F#L8EDP1"
28 H=0:V=0:IFPEEK(341)=247THENH=
0:V=-32ELSEIFPEEK(342)=247THENH=
0:V=32:ELSEIFPEEK(343)=247THENH=
-32:V=0:ELSEIFPEEK(344)=247THENH
=32:V=0
29 IFG1=1THENX1=-1:Y1=-1
30 IFG2=1THENX2=-1:Y2=-1
31 IFY=0ANDV=-32GOSUB80
32 IFY=32*5ANDV=32GOSUB80
33 IFX=0ANDH=-32GOSUB80
34 IFX=6*32ANDH=32GOSUB80
35 IFX1=X AND Y1=Y ANDG1<>1 THEN
63 ELSE IF X2=X AND Y2=Y ANDG2<
>1 THEN 63
36 IF TIMER>=40*60 THEN 63
37 IF TIMER>=30*60 THEN PLAY"T25
5;O3;1;7"
38 A=X+16+32*SGN(H):B=16+Y+32*SG
N(V)
39 IFPOINT(A,B)=3 GOSUB56
40 IFPOINT(X+H+16*SGN(X),Y+V+16
*SGN(Y))=2 ANDH<>0 AND V<>0THEN
63
41 IFX+H>=200ORH+X<0ORV+Y>=180OR
V+Y<0THENH=0:V=0
42 LINE(X,Y)-(X+32,Y+32),PRESET,
BF:X=X+H:Y=Y+V:PUT(X,Y)-(X+32,Y+
32),B
43 IFG1=1ANDG2=1THEN74
44 IF F1<0 THENF1=F1+1:GOTO50
45 IFG1=1THEN50ELSEIFH1=-1THENH1
=0:GOTO28ELSEH1=(RND(3)-2)*32:V1
=(RND(3)-2)*32:H2=(RND(3)-2)*32:
V2=(RND(3)-2)*32
46 IFX1+H1>=200ORX1+H1<0ORV1+Y1=

```

```

>1900RV1+Y1<9THEN50
47 IFPPOINT(X1+16+32*SGN(H1),16+
Y1+32*SGN(V1))=3 THENF1=-4
48 LINE(X1,Y1)-(X1+32,Y1+32),PRE
SET,BF:X1=X1+H1:Y1=Y1+V1:PUT(X1,
Y1)-(X1+32,Y1+32),BG
49 IFX1=X ANDY1=Y THEN63
50 H2=(RND(3)-2)*32:V2=(RND(3)-2
)*32:IFG2=1THEN28ELSEIFF2<0THENF
2=F2+1:GOTO28
51 IFX2+H2>=200ORX2+H2<0ORV2+Y2>
=32*6ORV2+Y2<0THEN28
52 IFPPOINT(X2+16+32*SGN(H2),16+
Y2+32*SGN(V2))=3THENF2=0
53 LINE(X2,Y2)-(X2+32,Y2+32),PRE
SET,BF:X2=X2+H2:Y2=Y2+V2:PUT(X2,
Y2)-(X2+32,Y2+32),BG
54 IFX2=X ANDY2=Y THEN63
55 GOTO 28
56 SC=SC+20:PLAY"T255;V31;01;1;3
;;04;1;5;3;2;4":XB=X+H:YB=Y+V
57 IFYB=0ANDH=0THENRETURN
58 IFYB=>32*5ANDV=32THENRETURN
59 IFXB=0ANDH=-32THENRETURN
60 IFXB=>32*6ANDH=32THEN RETURN
61 I=PPOINT(XB+16+32*SGN(H),YB+1
6+32*SGN(V)):IFI=3THEN RETURN:EL
SEIFI=2THEN 67
62 LINE(XB,YB)-(XB+30,YB+30),PRE
SET,BF:XB=XB+H:YB=YB+V:PUT(XB,YB
+4)-(XB+30,YB+30),BL:GOTO57
63 PLAY"01T255;1;1;1;1;1;2;2;2;2
;3;3;3;4;4;4;5;5;6;6;7;8;9;12;02
;1;4;7;9;12;03;1;5;8;12;04;1;6;1
2;05;1;12;T3;01;L4;DD;L8;D;L3;D;
L4;F;L8;E;L4;E;L8;D;L4;D;L8CL2D"
:CLS:PRINT"SCORE : "SC:IFSC=>H
S THEN HS=SC
64 PRINT:PRINT:PRINT"HIGH SCORE
: "HS:PRINT:PRINT:PRINT"LEVELS C
OMPLETED : "LE
65 A$=INKEY$:PRINT"PRESS ANY KEY
TO PLAY AGAIN"
66 PLAY"T255;01;1;2;3;4;1;2;3;4;
1;2;3;4;1;2;3;4;5;6;7;8;9;8;7;6;
5;4;3;2;1;2;3;4;5;6;7;6;5;6;7;6;
5;6;7;8;9;7;5;3;1;1;1":IFINKEY$=
""THEN66ELSESC=0:G1=0:G2=0:LE=0:
GOTO6
67 SC=SC+200:A=XB+32*SGN(H):B=YB
+32*SGN(V)
68 IFA=X1 AND B=Y1 THEN G1=1:ELS
EG2=1
69 IFA=X1 ANDB=Y1 THENPUT(X1,Y1)
-(X1+32,Y1+32),SG:ELSEPUT(X2,Y2)
-(X2+32,Y2+32),SG
70 PLAY"T255V3101;1;1;1;2;2;2;3;
3;3;4;4;4;5;5;6;6;7;8;8;9;10;11;
12;02;1;2;3;4;5;6;7;8;9;10;11;12
;03;1;3;5;7;9;11;12;04;1;5;9;12;

```

```

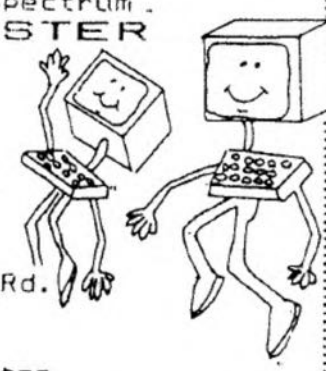
05;1;12"
71 LINE(A,B)-(A+32,B+32),PRESET,
BF:PUT(A,B+4)-(A+30,B+30),BL
72 LINE(XB,YB)-(XB+32,YB+32),PRE
SET,BF
73 GOTO28
74 T=TIMER:PLAY"T255;05;12;12;12
;12;10;10;10;9;9;9;8;8;8;7;7;7;6
;;6;6;5;4;3;2;1;04;12;10;9;8;7;
6;5;4;3;2;1;03;12;10;8;6;4;2;1;0
2;12;8;4;1;01;12;1"
75 CLS:PRINT@0,SC:PRINT@32*8,"LE
VELS COMPLETED : "LE+1:LE=LE+1:F
ORI=1TO40-T/60:T=T+60:SC=SC+10:P
RINT@128,"TIME LEFT:"INT(40-T/60
):PRINT@0,"SCORE: "SC:PLAY"T
25502;1;3;4;3;6;7;7;8":NEXT
76 PLAY"P4;02T3;L8;BBBABAB03D02B
L4ABPBLBBB BBBBAGAGL4BP4PBLBBAGED
L4ELBEEF#EDD1B02L4DEL8GEP2L4ELBF
#D"
77 A$=INKEY$:PRINT@360,"PRESS AN
Y KEY"
78 IFINKEY$=""THEN78
79 SCREEN1,0:G1=0:G2=0:SC=SC+500
:TIMER=0:GOTO 20
80 PLAY"T25502;1;3;5;7;9;7;5;3;1
;3;5;7;9;7;5;3;1;3;5;7;9;7;5;3;1
;1;1;3;4;1;2;3;4;3;2;1;2;3;4;1;1
;1;1;1":IFH=0THEN84
81 IFX1=X THENF1=-10:PUT(X1,Y1)-
(X1+32,Y1+32),SG
82 IFX2=X THENF2=-5:PUT(X2,Y2)-
(X2+32,Y2+32),SG
83 GOTO41
84 IFY1=Y THENF1=-10:PUT(X1,Y1)-
(X1+32,Y1+32),SG
85 IFY2=Y THENF2=-5:PUT(X2,Y2)-
(X2+32,Y2+32),SG
86 GOTO41

```

Software Spectrum
DONCASTER

- Adventures
- Arcades
- Business
- Word Processors
- Utilities

1/234 Blackburn Rd.
East Doncaster
Melb. Vic. 3109.
Phone (03) 842-1205





We mean business...

As with Education, this month we have suspended the series on the the programs we use here, to present a pair of business programs from the USA. We think they may provide a good kick off point for improved versions that suit the way we do things here.

Kevin has recently overcome some of his initial problems with his eighty column card, and so he has been proudly flaunting it and it's capabilities at every opportunity.

I wouldn't tell him this, but I must admit - it looks good - especially so with Stylograph.

The eighty column card puts CoCo right in the big league of business computers, because when you add CoCo's other abilities with the wider screen format, you get a most capable computer. It certainly eats the Model 4. May be that's why Tandy doesn't like seeing CoCo's used for business!

We have a copy of DynaCalc here, an advanced spreadsheet, which Sheryl is wading through, preparatory to reviewing it in the magazine. If you have only ever seen Spectaculator, then I think you may be surprised by the power of DynaCalc. (Not this withstanding, my opinion on spreadsheets has not changed!)

FINANCE

16K
ECB

the
RAINBOW

Watching Your Cash Flow Rate of Return

By William S. Bonnell

In business, projects are attractive if they produce a rate of return on investment greater than a company's cost of capital. This produces a net inflow of cash into the company. If a company has unlimited funds, it invests in all projects having a rate of return greater than the company cost of capital. In reality, projects are rejected even though they have a favorable rates of return because funds are limited.

Capital budgeting ranks projects according to their financial attractiveness and selects the set which maximizes the present worth (PW) of the firm. One method of capital budgeting compares projects on the basis of present worth, considering those for investment which have a cash flow rate of return (CFRR) greater than some minimum required rate of return, the cost of capital.

The program described in this article determines the CFRR and PW of a project and allows sensitivity testing of the assumptions made in the calculations.

The required parameters for calculation of CFRR and PW for a project are:

- 1) Expenses by year
- 2) Savings by year
- 3) Capital investment by year
- 4) Tax rate percent
- 5) Investment tax credit percent
- 6) Type of depreciation
- 7) Depreciation life of capital
- 8) The required rate of return percent

Expenses are defined as the annual cost of operations. They are not depreciated. They represent the outflow of cash resulting from the alternative being investigated. Savings are the opposite of expenses and represent the inflow of cash from the alternative in question. Expenses can be netted out of savings or both can be entered separately. Examples of expenses are labor, overhead and raw materials. Savings result from prevented expenditures or reductions in labor, overhead and raw materials.

Capital expenditures are investments in building and equipment which are allowed by law to be recovered over time through depreciation.

The tax rate is the percentage of
AUSTRALIAN RAINBOW



profits which the organization must pay to the government. Investment tax credit is a means the government uses to stimulate business investment in capital equipment. Generally in the year following the capital expenditure, the organization is not taxed for a percentage of the expenditure.

The required rate of return depends on the nature of the company involved.

If the company finances its operations by borrowing, the cost of capital is the borrowing rate. If projects have a rate of return greater than the borrowing rate, they will be profitable. Therefore, the required rate of return is the borrowing rate.

In companies financing their operations by common stock or a combination of stock and debt, the cost of capital is more difficult to determine. In these cases, the cost of capital may be a policy variable supplied by management as a benchmark for comparison of projects in general. It may be unrelated to the true cost of capital funds to the company.

The PW of a project is defined as

the net cash flow (inflow-outflow) when all flows have been discounted to the present. The basis for this concept is a dollar invested for one year is worth more than a dollar at the end of the year. Similarly, a dollar received a year from now is not worth as much as a dollar now. The value now of a dollar received a year from now is:

$$1/(1+r)$$

where 'r' is the rate of return a dollar could earn if invested at the company's cost of capital. The value now of one dollar received two years from now is:

$$1/(1+r)**2$$

Note: ** is used as a symbol for exponentiation.)

And in general, the value of a dollar received N years from now is:

$$1/(1+r)**N$$

If a company spends \$1,000 on a piece of equipment which will save \$400 each year for 5 years, what is the PW of this project, ignoring taxes and depreciation, if the company can earn 10% from other investments?

$$-1000/(1+.1)**0 + 400/(1+.1)**1 + 400/(1+.1)**2 + 400/(1+.1)**3 + 400/(1+.1)**4 + 400/(1+.1)**5$$

The present worth (PW) is \$516.31. The straightforward answer might seem to be \$1,000 since \$1,000 went out and \$2,000 came in, but this ignores the alternative uses of these funds.

Cash flow rate of return (CFRR) is defined as the rate, 'r,' which makes PW equal to zero in the following equation:

$$PW = CF1 + CF2/(1+r)**1 + CF3/(1+r)**2 + \dots + CFN/(1+r)**N-1$$

where "CFN" is the net cash flow (in-out) at the beginning of year 'N' (or at the end of year N-1).

CFRR is analogous to lending a sum of money and receiving annual payments which eventually pay off the original deposit plus interest. The interest rate earned is like CFRR.

To be accurate and of use in business, tax and depreciation effects on taxes must be considered. Depreciation is a method for recovering the cost of capital assets over time. Government policy determines how quickly the investment can be recovered. Faster recovery is more beneficial to business because of the time-value of money.

Money recovered sooner is more valuable than money recovered later.

There are many methods of depreciation depending on the type and life of the asset. Using straight line depreciation, 20% of an asset's value can be recovered each year after the first, assuming a five-year tax life. It is recovered by subtracting depreciation from profits. This means a company can shield a portion of its profits from taxes by depreciating its assets. This is an incentive for investment in new assets.

An accelerated method of depreciation is Sum of the Year's Digits (SOYD). If an asset has a depreciation life of five years, the SOYD is 15 (1+2+3+4+5). In the first year after the capital expenditure, this method would allow $\frac{1}{15}$ or .3333 of the asset value to be deducted from profits before calculating taxes. In subsequent years $\frac{2}{15}$, $\frac{3}{15}$, $\frac{4}{15}$, $\frac{5}{15}$ of the asset value could shield profits from taxes. The sum of these depreciation factors is usually one.

Double-declining balance is another accelerated method of depreciation. In the first year after capital expenditure, the depreciation factor is two divided by the asset life. In subsequent years, the remaining fraction to be depreciated is multiplied by 2/(asset life). The remaining fraction is the previous remaining fraction less the current depreciation factor. Refer to Listing 1, lines 670-1100 for calculation of depreciation factors.

Once the depreciation factors have been determined, the after-tax cash flow is calculated for each year.

$$\text{Cash Flow} = \text{Savings} - \text{Tax Rate} * (\text{Tax Base}) - (\text{Expense} + \text{Capital}) + \text{Investment Tax Credit}$$

$$\text{Tax Base} = \text{Savings} - (\text{Expense} + \text{Depreciation})$$

$$\text{Investment Tax Credit} = \text{Tax Credit \%} * \text{Capital Investment In The Previous Year}$$

Refer to Listing 1, lines 1110 - 1250 for this calculation.

This annual net cash flow is then adjusted for the time-value of money at the company cost of capital. It is summed to get the net present worth of the project. The CFRR is calculated by finding rate of return, 'r,' which just makes cash inflows equal to cash outflows, causing PW to equal zero.

The Program

The program was written for the 16K Color Computer using disk or cassette.

It should run on any computer using standard BASIC with very few modifications.

Data entry is prompted or menu driven. If flows become constant, the program allows one value to be entered for several years. There are four standard types of depreciation to choose from or factors may be entered manually.

After the data has been entered, the program calculates and displays the depreciation flows, the after-tax cash flow and the discounted after-tax cash flow using 15% as the discount rate. Present worth is calculated for various required rates of return. When PW changes sign from a positive to negative (passes through zero), that is the CFRR.

After the display of output is complete, data values can be changed and the problem rerun. This is a powerful tool for sensitivity analysis. Finally, data may be saved or read from disk or cassette. A report can be generated on a printer.

The algorithm to calculate CFRR is shown in Listing 1, lines 1510 to 1790. It is a binary search algorithm. If CFRR is in the range zero to 100 percent, the interval containing the CFRR is halved each iteration until one endpoint is within three percent of the other.

At that point, the program interpolates for a final value of CFRR. For example, if the PW at R=100% is less than zero and PW at R=0 is greater than zero, PW is calculated at R=50%. If PW at 50 percent is less than zero, PW is found at R=75%. If PW at R=75% is greater than zero then PW is found at R=62.5%, and so on until the difference between successive R's is less than three percent. The program then interpolates the final CFRR.

To illustrate its use and verify your entry of the program, try the following example problem:

A company spends \$10,000 on a new computer system. By spending this amount, labor savings of \$4,000 are expected each subsequent year for 10 years. The company tax rate is 48% and an investment tax credit of 10% is available. It will cost \$500 per year for supplies for the new computer and \$400 per year for maintenance.

Computers can be depreciated with a five year life using double-declining balance switching to SOYD. What is the present worth of this investment if the company can earn 15% on alternative investments? What is the CFRR for this project?

INPUT DATA

Start Year	Expense	Savings	Capital
1	0	0	10000
2 - 11	900	4000	0

Project life=11 years
 Depreciation life=5 years
 Tax rate=48 percent
 Investment tax credit=10 percent

Required Rate Of Return = 15%
 Tax Rate = 48%
 Investment Tax Credit = 10%

Listing 2 shows my inputs and the computer responses to solve this problem. It was produced by echoing the screen output to the printer. A printer report is also generated.

At 15 percent, this project has a positive net present worth indicating a worthwhile investment. The CFRR of 22 percent is greater than can be earned on the average investment available to the company (15 percent).

If other investments are available, they should be compared on the basis of present worth. Comparing on the basis of CFRR can, in some cases, lead to the wrong conclusion. The CFRR should be a "go/no go" comparison with the required rate of return for the company.

Hopefully, the value of this program has been demonstrated for analyzing alternative business investments. The concept is more fully explained in any good financial textbook.

Year	Expense	Savings	Capital	Dpr. Flow	Cash Flow
1	\$ 0.00	\$ 0.00	\$10,000.00	\$ 0.00	-\$10,000.00
2	900.00	4,000.00	0.00	2,000.00	3,572.00
3	900.00	4,000.00	0.00	3,200.00	3,148.00
4	900.00	4,000.00	0.00	2,100.00	2,620.00
5	900.00	4,000.00	0.00	1,500.00	2,332.00
6	900.00	4,000.00	0.00	900.00	2,044.00
7	900.00	4,000.00	0.00	300.00	1,756.00
8	900.00	4,000.00	0.00	0.00	1,612.00
9	900.00	4,000.00	0.00	0.00	1,612.00
10	900.00	4,000.00	0.00	0.00	1,612.00
11	900.00	4,000.00	0.00	0.00	1,612.00

Required	Present Worth
15%	-\$2307.51
30%	- 1742.79
45%	- 3846.28
60%	- 5107.03
75%	- 5940.91
90%	- 6532.03

CFRR = 22.18%

List of Variables

Variable	Function
A	Choice of option
AS	Answer to prompts
CA()	Capital expenditure array
CC	Capital input value
CF()	Net cash flow array
D	Device for I O
DC	Discounted cash flow
DF()	Depreciation factors
DL	Depreciation life
DP()	Depreciation component of cash flow
DT	Depreciation type
EI	End year for editing changes
EE	Expense input
EX()	Expense array
FS	Filename

FR	Double-declining balance fraction	RR	depreciation Required rate of return (ROR)
I,II,		RR()	RORs used to calculate PW()
J,K	Various year counters	RV	Remaining value in depreciation
N	Net present value (NPV)	SI	Start year for editing data
N6	NPV in CFRR interpolation	SA()	Savings array by year
OD	Old depreciation life	SF	Sum of cash flows in finding payback
OP	Old project life	SS	Savings input value
P9	Payback time	SY	Sum of years digits
PB	Discounted payback time	TC	Investment tax credit value
PL	Project life	TX	Tax rate
PW	Present worth (PW)	VI	Value to change in editing data
PW()	PW array for various rates of return	YY	Year when input flows become constant
R	Calculated CFRR		
R3,R4,			
R5,R6	Used in binary search CFRR calculation		
RI	Remaining life in		

19048	175025
430122	200026
6704	2250104
97033	24905
122058	2680118
1470241	END63

```

40 PRINT " * 3/24/1983 COPYRI
GHT * "
50 PRINT " *****
***** "
60
70 PRINT:PRINT
80 DIM EX(20),SA(20),CA(20),DF(2
0),DP(20),CF(20),RR(6),PW(6)
90 INPUT "MENU ENTRY OR REGULAR
ENTRY M/R";A$
100 IF LEFT$(A$,1)="M" THEN 1010
110 GOSUB 130:GOSUB 190:GOSUB 36
0:GOSUB430:GOSUB500:GOTO 1110
    
```

```

10 CLS
20 PRINT " *****
***** "
30 PRINT " * CFRR PROGRAM, W. BO
NNELL * "
PAGE 20
    
```

```

120 '
130 IF PL>0 THEN CLS:PRINT"OLD P
ROJECT LIFE="OP"YEARS"
140 INPUT "PROJECT LIFE";PL:OP=PL
L
150 IF PL>20 THEN 130
160 IF PL<1 THEN 130
170 RETURN
180 '
190 IF OD>0 THEN CLS: PRINT"OLD
DEPRECIATION LIFE="OD
200 INPUT "DEPRECIATION LIFE";DL
:OD=DL
210 CLS
220 IF DL>20 THEN 190
230 IF DL<1 THEN 190
240 PRINT
250 PRINT "DEPRECIATION TYPES:"
260 PRINT "1. SUM OF YEARS DIGIT
S"
270 PRINT "2. DOUBLE DECLINING B
ALANCE"
280 PRINT "3. STRAIGHT LINE"
290 PRINT "4. DDB->SOYD 1/2YR.AV
ERAGING"
300 PRINT "5. ENTER OWN FACTORS"
310 INPUT"CHOOSE";DT
320 ON DT GOSUB 690,740,820,900,
1030
330 PRINT
340 RETURN
350 '
360 IF TX>0 THEN CLS:PRINT"OLD T
AX RATE=";TX*100
370 INPUT "TAX RATE IN %";TX
380 IF TX>99 THEN 360
390 IF TX<0 THEN 360
400 TX=TX/100
410 RETURN
420 '
430 IF TC>0 THEN CLS: PRINT "OLD
INV. TAX CREDIT=";TC*100
440 INPUT "INV. TAX CREDIT IN %"
;TC
450 IF TC<0 THEN 430
460 IF TC>99 THEN 430
470 TC=TC/100
480 RETURN
490 '
500 INPUT"DO ANNUAL FLOWS BECOME
CONSTANT Y/N ";A$
510 IF LEFT$(A$,1)<>"Y" THEN 590
520 CLS
530 INPUT "YEAR WHEN FLOWS BECOM
E CONSTANT";YY:IF YY>PL THEN 530
540 PRINT "FOR YEAR WHEN FLOWS A
RE CONSTANT ENTER:"
550 PRINT
560 PRINT "EXPENSE,SAVINGS,CAPIT
AL FLOWS": INPUT EE,SS,CC
570 FOR I=YY TO PL:EX(I)=EE:SA(I
)=SS:CA(I)=CC:NEXT I
580 PRINT
590 PRINT"YEAR EXPENSE, SAVING,
CAPITAL"
600 IF YY=0 THEN YY=PL
610 FOR J=1 TO YY-1
620 PRINT USING "## ";J;
630 INPUT EX(J),SA(J),CA(J)
640 NEXTJ
650 RETURN
660 '
670 'GET DEPRECIATION FACTORS
680 'SUM OF YEARS DIGITS TYPE
690 IF INT(DL)=DL THEN SF=DL*(DL
+1)/2 ELSE SF=(DL+.5)*(DL+.5)/2
700 FOR J=DL TO 0 STEP -1
710 DF(DL-J+1)=J/SF
720 NEXT J
730 RETURN
740 'DOUBLE DELCINING BALANCE
750 RV=1
760 FOR I=1 TO DL+.5
770 FR=2/DL
780 DF(I)=RV*FR
790 RV=RV-DF(I)
800 NEXT I
810 RETURN
820 'STRAIGHT LINE
830 RV=1
840 FOR I=1 TO DL
850 DF(I)=1/DL
860 RV=RV-DF(I)
870 NEXT I
880 DF(I)=RV
890 RETURN
900 'DDB TO SOYD 1/2 YR AVG CONV
NTN SWITCH IN 2ND YEAR
910 DF(1)=1/DL
920 RV=1-DF(1)
930 DF(2)=2/DL*RV
940 RV=RV-DF(2)
950 RL=DL-1.5
960 IF INT(RL)=RL THEN SY=RL*(RL
+1)/2:EN=1 ELSE SY=(RL+.5)*(RL+.
5)/2
970 K=2
980 FOR I=RL TO 0 STEP -1
990 K=K+1
1000 DF(K)=I/SY*RV
1010 NEXT
1020 RETURN
1030 'OWN FACTORS
1040 CLS:PRINT "ENTER"DL"FACTORS
"
1050 FOR I=1 TO DL
1060 PRINT USING "## ";I;
1070 INPUT DF(I)
1080 NEXT
1090 RETURN

```

```

1100
1110 GET DEPR. AND AFTER TAX FLOW
OWS
1120 CLS
1130 PRINT:PRINT "START DEPREC A
FTER TAX DISCOUNT"
1140 PRINT"OF YR. FLOW CASH FLO
W @ 15%"
1150 FOR I=1 TO PL
1160 IF I>DL THEN I1=I-INT(DL) E
LSE I1=1
1170 FOR K=I1 TO I
1180 DP(I)=DP(I)+CA(K-1)*DF(I-K+
1)
1190 NEXT K
1200 CF(I)=SA(I)-TX*(SA(I)-(EX(I
)+DP(I)))-(EX(I)+CA(I))+TC*CA(I-
1)
1210 PRINT USING "## ";I;
1220 PRINT USING "#####.##";DP(
I);CF(I);CF(I)/(1.15)^(I-1)
1230 NEXT I
1240 INPUT "HIT ENTER TO CONTINU
E";A$
1250 CLS
1260
1270 PRESENT WORTH
1280 PRINT
1290 PRINT "REQUIRED AFTER TAX
"
1300 PRINT " RATE % PRESENT WOR
TH"
1310 J=0
1320 FOR R=.15 TO .91 STEP .15
1330 J=J+1:RR(J)=R*100
1340 PW=0:SM=0
1350 FOR I=1 TO PL
1360 DC=CF(I)/(1+R)^(I-1)
1370 SM=SM+CF(I)
1380 PW=PW+DC
1390 IF PB=0 THEN IF PW>0 THEN P
B=I:RR=R
1400 IF P9=0 THEN IF SM>0 THEN P
9=I
1410 NEXT I
1420 PRINT USING " ## % ";100*
R;
1430 PRINT USING "#####.##";PW
1440 PW(J)=PW
1450 NEXT R
1460 IF P9-1>0 THENPRINT "STRAIG
HT PAYBACK"P9-1"YEARS"
1470 IF PB-1>0 THENPRINT "DISCOU
NTED PAYBACK"PB-1"YEARS" ELSE PR
INT "NO DISCOUNTED PAYBACK"
1480 IF RR=0 THEN RR=.15
1490 PRINT " AT "RR*100"%
1500
1510 GET CFRR
1520 R=0:R3=0:R5=0:R6=0:N6=0

```

```

1530 R4=1
1540 GOSUB 1690
1550 IF N<=0 THEN GOTO 1760
1560 R=R4
1570 GOSUB 1690
1580 IF N>=0 THEN R=100:GOTO 176
0
1590 R=(R3+R4)/2
1600 IF ABS(R6-R)<=.03 THEN 1730
1610 R6=R:N6=N
1620 GOSUB 1690
1630 IF N<0 THEN GOTO 1660
1640 IF N>0 THEN 1670
1650 IF N=0 THEN 1760
1660 R4=R:GOTO 1590
1670 R3=R:GOTO 1590
1690 PRINT@416,"wait ";;N=0:FOR
I=1 TO PL:N=N+CF(I)/(1+R)^(I-1):
NEXT I
1700 PRINT @420,STRING$(4,8);
1710 RETURN
1720 GOTO 1780
1730 N6=N
1740 GOSUB 1690
1750 R=(R-N*(R-R6)/(N-N6))*100
1760 R=INT(100*R+.5)/100
1770 PRINT
1780 PRINT@416,"CFRR="R%"
1790 INPUT "HIT ENTER TO CONTINU
E";A$:PRINT
1800
1810 CHANGE DATA ROUTINE
1820 CLS
1830 PRINT " entry/change menu
"
1840 PRINT "1. PROJECT LIFE"
1850 PRINT "2. DEPRECIATION LIFE
"
1855 PRINT "3. EXPENSES"
1860 PRINT "4. SAVINGS"
1870 PRINT "5. CAPITAL"
1890 PRINT "6. TAX RATE"
1900 PRINT "7. INV. TAX CREDIT"
1910 PRINT "8. SAVE DATA IN FILE
"
1920 PRINT "9. READ DATA IN FILE
"
1930 PRINT "10.RUN"
1940 PRINT "11.PRINT REPORT-(AFT
ER OPT.10)"
1950 PRINT "12.STOP"
1960 INPUT "CHOOSE";A
1970 ON A GOSUB 130,190,2020,212
0,2220,360,430,2320,2420,2540,26
20
1980 IF A=10THEN GOSUB 320:GOTO
1120
1990 IF A=12THEN STOP
2000 GOTO 1810
2010
2020 CLS:PRINT"YR

```



```

YR      EXPENSE"
2030 FOR I=1 TO PL
2040 PRINT USING "## ";I;:PRINT
USING "$$#####.## ";EX(I);
2050 NEXT
2060 PRINT:PRINT"ENTER START,END
YR,EXPENSE":INPUT S1,E1,V1
2070 FOR I=S1 TO E1
2080 EX(I)=V1
2090 NEXT
2100 RETURN
2110 '
2120 CLS:PRINT"YR      SAVINGS
YR      SAVINGS"
2130 FOR I=1 TO PL
2140 PRINT USING "## ";I;:PRINT
USING "$$#####.## ";SA(I);
2150 NEXT
2160 PRINT:PRINT"ENTER START,END
YR,SAVINGS":INPUT S1,E1,V1
2170 FOR I=S1 TO E1
2180 SA(I)=V1
2190 NEXT
2200 RETURN
2210 '
2220 CLS:PRINT"YR      CAPITAL
YR      CAPITAL"
2230 FOR I=1 TO PL
2240 PRINT USING "## ";I;:PRINT
USING "$$#####.## ";CA(I);
2250 NEXT
2260 PRINT:PRINT"ENTER START,END
YR,CAPITAL":INPUT S1,E1,V1
2270 FOR I=S1 TO E1
2280 CA(I)=V1
2290 NEXT
2300 RETURN
2310 '
2320 INPUT"ENTER OUTPUT FILE NAM
E";F$
2330 INPUT"TO DISK OR TAPE D/T";
A$: IF LEFT$(A$,1)="D" THEN D=1 E
LSE D=-1:AUDIOON
2340 OPEN "O",#D,F$
2350 PRINT #D,PL;DL;TX;TC;DT;OD;
OP;
2360 FOR I=1 TO PL
2370 PRINT #D,EX(I);SA(I);CA(I);
2380 NEXT I
2390 CLOSE D
2400 RETURN
2410 '
2420 INPUT"ENTER INPUT FILE NAME
";F$
2430 INPUT"FROM DISK OR TAPE D/T
";A$: IF LEFT$(A$,1)="D" THEN D=1
ELSE D=-1:AUDIOON
2440 OPEN "I",#D,F$
2450 IF EOF(D)=-1THEN GOTO2500
2460 INPUT #D,PL,DL,DX,TC,DT,OD,
OP
2470 FOR I=1 TO 20
2480 IF EOF(D)=-1THEN I=20:GOTO2
500
2490 INPUT #D,EX(I),SA(I),CA(I)
2500 NEXT I
2510 CLOSE D
2520 RETURN
2530 '
2540 'INITIALIZE THEN RERUN
2550 FOR I=0 TO 20
2560 DF(I)=0:CF(I)=0:DP(I)=0
2570 NEXT
2580 FB=0:F9=0:RR=0
2590 GOSUB 320 'DEPREC
2600 GOTO 1120
2610 '
2620 'PRINTER REPORT
2630 PRINT#-2,"*****
*****"
2640 PRINT#-2,"* CFRR PROGRAM, W
. BONNELL *"
2650 PRINT#-2,"*      3/24/1983 CO
FYRIGHT  *"
2660 PRINT#-2,"*****
*****"
2670 PRINT#-2:PRINT#-2
2680 PRINT#-2, "PROJECT LIFE="
OF"YEARS"
2690 PRINT#-2,"DEPRECIATION LIFE
="OD"YEARS"
2700 PRINT#-2,"TAX RATE="TX*100"
%"
2710 PRINT#-2,"INVESTMENT TAX CR
EDIT="TC*100%"
2720 PRINT#-2
2730 PRINT #-2,"YEAR      ";
2740 PRINT#-2,USING "%      %
";" EXPENSE";" SAVINGS";" CAP
ITAL";"DPR. FLOW";"CASH FLOW"
2750 FOR I=1 TO PL
2760 PRINT#-2,USING "##      ";I;
2770 PRINT#-2,USING "$$#####.##
";EX(I);SA(I);CA(I);DP(I);CF(I)
2780 NEXT I
2790 PRINT#-2
2800 PRINT#-2,"REQUIRED PRESEN
T WORTH"
2810 FOR I=1 TO 6
2820 PRINT #-2,USING"      ## %
";RR(I);
2830 PRINT#-2,USING "$$#####.##
# ";PW(I)
2840 NEXT I
2850 PRINT #-2
2860 PRINT #-2,"CFRR="R"%
2870 RETURN

```

In making the most of your time, this program could be the solution

PERT: Project Evaluation and Review Technique

By Jorge Mir

Having the ability to evaluate and review projects can be very useful, at the office as well as at home, especially when time is of the essence and you want to make sure that projects are completed on time with a minimum of supervision.

There are many types of "PERT" programs available on the market. Some of these are quite sophisticated (and expensive) and require a large mainframe in which to operate. The PERT version I have developed for the Color Computer is a simple one, yet it has sufficient "bells and whistles" to make it quite beneficial at work.

Following are the main features of the PERT program:

- 1) Calculates minimum time needed to complete project.
- 2) Calculates probability of completion by a target date.
- 3) Determines whether activities are critical or non-critical.
- 4) Calculates expected duration and standard deviation for each activity.
- 5) Determines early and late start and finish times.
- 6) Calculates slack time for each activity (if any).

In addition, several output reports

are available:

- 1) Worksheet for developing a precedence table.
- 2) Complete activity schedule.
- 3) Activity schedule by responsibility.
- 4) Complete Gantt type schedule.
- 5) Gantt schedule by responsibility.

The activity and Gantt schedules both clearly indicate the "critical" activities so the user can concentrate on evaluating those activities, as well as monitoring progress, as the project advances through the various stages of completion.

The program is fully prompted for all input data, so these will not be covered in detail. The input and output routines send data to a disk. If you have a cassette system, you will have to change those routines so data is saved to cassettes instead of disks. Also, a printer is necessary for the various output reports.

Please note that a total of 100 activities is the maximum established by the program. This limitation can be changed by changing the DIM statements at the beginning of the program.

In order to properly enter all data, a precedence table (which activity comes before another activity) is necessary. After you enter all of the activities, you will be able to develop a worksheet for this purpose. After you have the worksheet completed, you can then run the program in its entirety.

Here are some hints for properly completing the worksheet:

A "node" can be viewed as a stage in the project. That is, at which stage an activity must commence and at which stage it must end. These nodes serve as the key for determining the rest of the calculations in the program, so care should be exercised in planning this aspect of the project.

The start node of each activity must appear as the end node of some other activity in the project. In other words, all activities must be linked with each other (exceptions are those activities with a node of '1', meaning they are to be performed first and no other activity precedes it).

More than one activity can have the same start or end nodes; just make sure the beginning node of an activity coincides with the ending node of the activity that must precede it. Here is a simple example (cooking a meal):

	Start Node	End Node
Buy all ingredients	1	2
Marinate meat	2	3
Wash vegetables	2	4
Bake potatoes	3	5
Broil meat	3	6
Cook vegetables	4	6
Serve meal	6	7

In the example, you cannot broil the meat (start node 3) until the meat is marinated (end node 3). Also, this project may require two people since there are two activities starting at the same node (i.e., one marinating the

meat and the other washing the vegetables).

If there is a start node which does not coincide with an ending node in one of the other activities, the program will stop execution and inform you of those activities for which starting nodes must

be changed.

The best way to get used to the program is to run some simple examples, as the one noted. That way, you will get a good feel for all the prompts for inputting data and all of the output reports, as well as the various ways in

which a program of this nature can be put to good use.

I often get involved in some complex projects at work and have used this program many times to help me keep control of such projects. Take my word for it, it really helps!

The listing:

```

10 GOTO 2680
20 CLEAR 10000
30 DIM A$(100),R$(100),A(100,2),
S(100),F(100),E(100,2),C(100)
40 DEFFNR(Z1)=INT((Z1*1000+.5))/
1000
50 GOTO 1530
60 GOSUB1540:PRINT"HAVE ACTIVITI
ES ALREADY BEEN STORED ON DIS
K?"
70 I$=INKEY$:IF I$=""THEN70 ELSE
IF I$<>"Y" THEN PT=0:GOTO150 EL
SE PT=1
80 GOSUB1540:PRINT"PLACE DISK CO
NTAINING PREVIOUSLYSTORED ACTIVI
TIES IN DRIVE 0 ANDPRESS ANY KEY
WHEN READY."
90 IF INKEY$=""THEN90 ELSE PRINT
:PRINT"LOADING DATA....."
100 OPEN"I",#1,"PERT.ACT"
110 INPUT#1,PT$
120 IF EOF(1) THEN 140
130 N=N+1:INPUT#1, A$(N),R$(N):G
OTO120
140 CLOSE#1:GOTO250
150 GOSUB 1540:PRINT"TITLE OF TH
IS PROJECT:":LINE INPUT PT$
160 PRINT
170 PRINT"DESCRIBE ACTIVITIES IN
CLUDED IN THIS PROJECT."
180 PRINT"TYPE 'END' WHEN DONE."
:PRINT:A=PEEK(&H88)*256+PEEK(&H8
9)
190 PRINT@A-&H400,"ACTIVITY NO."
;N+1;CHR$(8)":":PRINT:PRINT:PRIN
T:PRINT@A-&H400+32,;:LINE INPUT
A$
200 IF A$="ERROR" THEN N=N-1:GOT
O 190
210 IF A$="END" THEN 250
220 N=N+1
230 I=INSTR(A$,"/"):IF I>0 THEN
A$(N)=LEFT$(A$,I-1):R$(N)=MID$(A
$,I+1) ELSE SOUND100,2:N=N-1:PRI
NT@A-&H400,"ENTER '/' PLUS RESPON
SIBILITY!":FOR Q=1TO500:NEXTQ:SO
UND100,2:GOTO190
240 GOTO 190
250 GOSUB 1540:PRINT"DO YOU WANT
TO PREPARE WORKSHEET TO DEVELOP
A PRECEDENCE TABLE?"

```

```

260 I$=INKEY$:IF I$=""THEN 260
270 IF I$<>"Y" THEN 520
280 IF PT=1 THEN PT=0:GOTO380
290 GOSUB 1540:PRINT"DATA MUST B
E STORED BEFORE WE PROCEED."
300 PRINT:PRINT"PRESS ANY KEY WH
EN YOU HAVE A DISK READY FOR S
TORING DATA."
310 IF INKEY$=""THEN310
320 OPEN"O",#1,"PERT.ACT"
330 WRITE#1,PT$
340 FOR X=1TON
350 WRITE#1,A$(X),R$(X)
360 NEXT X
370 CLOSE#1
380 GOSUB 1540
390 PRINT"PLEASE GET PRINTER REA
DY FOR PRINTING ACTIVITY WORK
SHEET."
400 PRINT:PRINT"PRESS ANY KEY WH
EN READY."
410 IF INKEY$=""THEN 410
420 PRINT#-2,"PROJECT: ",PT$
430 PRINT#-2:PRINT#-2
440 PRINT#-2,"
TIMES"
NODES
450 PRINT#-2,"
-----
-----"
460 PRINT#-2,"
ACTIVIT
IES START E
ND BEST PAR WORST"
470 PRINT#-2,"-----
-----"
480 FOR X=1TO N
490 PRINT#-2:PRINT#-2,USING"###
-%
% -----"
500 NEXT X
510 PT$="":GOTO 1530
520 '*** ENTER DATA ***
530 FOR I=1 TO N
540 IF A(I,1)>0 THEN 800
550 GOSUB1540
560 PRINT A$(I):PRINT
570 PRINT"START NODE";
580 INPUT A(I,1)
590 IF A(I,1)<0 THEN I=I+A(I,1):
GOTO550
600 PRINT"END NODE";

```

```

610 INPUT A(I,2)
620 IFA(I,2)>=A(I,1) THEN 680
630 IF A(I,2)<N THEN 680
640 PRINT " START NODE MUST BE NU
MBERED LOWER THAN END NODE,
AND END NODE MUST BE LESS THA
N NUMBER OF ACTIVITIES."
650 PRINT " *** TRY AGAIN ***"
660 PRINT
670 GOTO 550
680 PRINT"ENTER THREE TIME ESTIM
ATES FOR THIS ACTIVITY:"
690 PRINT " OPTIMISTIC TIME:";
700 INPUT A1
710 PRINT " EXPECTED TIME:";
720 INPUT M
730 PRINT " PESSIMISTIC TIME:";
740 INPUT B
750 E(I,1)=FNR((A1+M*4+B)/6)
760 E(I,2)=FNR((B-A1)/6)^2
770 S(I)=0
780 F(I)=0
790 IF FLAG=1 THEN FLAG=0:RETURN
800 NEXT I
810 FOR I=1 TO N
820 IF S(A(I,2))>=S(A(I,1))+E(I,
1) THEN 840
830 S(A(I,2))=S(A(I,1))+E(I,1)
840 NEXT I
850 F(A(N,2))=S(A(N,2))
860 FOR I = N TO 1 STEP -1
870 IF F(A(I,1))=0 THEN 900
880 IF F(A(I,1))>F(A(I,2))-E(I,1
) THEN 900
890 GOTO 930
900 IF F(A(I,2)) < S(A(I,1))+E(I
,1) THEN F(A(I,2))=S(A(I,1))+E(I
,1)
910 Q=F(A(I,2))-E(I,1):IF Q=>0 T
HEN F(A(I,1))=Q
920 IF F(A(I,1))=0 THEN F(A(I,1)
)=S(A(I,1))
930 NEXT I:GOTO 1430
940 V=0:U$="" %
- ###.###"
950 C=0
960 L=0
970 FOR I= 1 TO N:CLS
980 S1=F(A(I,2))-S(A(I,1))-E(I,1
):IF INT(S1)=<0 THEN C(I)=0 ELSE
C(I)=S1
990 CLS:IF LEN(PT$)=>32 THEN PRI
NT PT$ ELSE PRINT TAB((32-LEN(PT
$))/2)PT$
1000 PRINT STRING$(32,131);
1010 PRINT A$(I):PRINT
1020 PRINT"FROM NODE";A(I,1)"TO
NODE";A(I,2)
1030 PRINT "(A ";
1040 IF C(I)=0 THEN 1060

```

```

1050 PRINT"NON-";
1060 PRINT"CRITICAL EVENT)":PRIN
T
1070 PRINTUSING U$;"EXPECTED DUR
ATION",E(I,1):PRINTUSINGU$;"STD.
DEVIATION",SQR(E(I,2))
1080 IF INT(S1)>0 THEN 1150
1090 PRINTUSINGU$;"START NO LATE
R THAN",S(A(I,1))
1100 PRINTUSING U$;"MUST COMPLET
E BY",F(A(I,2))
1110 IF L>=F(A(I,2)) THEN 1130
1120 L=F(A(I,2))
1130 V=V+E(I,2)
1140 GOTO 1190
1150 PRINTUSING U$;"EARLY START"
,S(A(I,1)):PRINT USINGU$;"LATE S
TART",F(A(I,2))-E(I,1)
1160 PRINT USING U$;"EARLY FINIS
H",S(A(I,1))+E(I,1)
1170 PRINTUSINGU$;"LATE FINISH",
F(A(I,2))
1180 PRINT USING U$;"SLACK TIME"
,S1
1190 PRINT@448,STRING$(32,140);;
PRINT@484,"PRESS ANY KEY TO CONT
INUE";
1200 I$=INKEY$:IF I$=""THEN1200
1210 IF I$=CHR$(13) THEN 1530
1220 NEXT I
1230 GOSUB1540
1240 PRINT"THE CRITICAL PATH LEN
GTH IS: ";L
1250 P=SQR(V)
1260 PRINT"PLUS OR MINUS";:PRINT
USING" #.##";P
1270 PRINT"ENTER DESIRED COMPLET
ION TIME (0 TO END)";
1280 INPUT D
1290 IF D<=0 THEN 1530
1300 Y=(D-L)/P
1310 R=EXP(-(Y^2)/2)/2.506628274
6
1320 Z=Y
1330 Y=1/(1+.33267*ABS(Y))
1340 T=1-R*(.4361836*Y-.1201676*
Y^2+.937298*Y^3)
1350 IF Z>=0 THEN 1370
1360 T=1-T
1370 GOSUB1540
1380 PRINT
1390 PRINT USING "PROBABILITY OF
COMPLETION WITH DURATION OF ##
# IS ##.##";D,T
1400 PRINT
1410 GOTO 1270
1420 CLS:END
1430 PRINT:PRINT" EDITING STARTI
NG TIMES....."
1440 FOR I = 1 TO N

```

```

1450 IF S(A(I,1))>0 THEN 1490
1460 IF A(I,1)=1 THEN 1490
1470 IF XX=0 THEN GOSUB1540:PRIN
T"THESE ITEMS NEED EDITING:":XX=
1
1480 PRINT " ";A$(I)
1490 NEXT I:XX=0
1500 PRINT@480," <PRESS ANY KEY
TO CONTINUE>";
1510 IF INKEY$=""THEN1510
1520 GOTO 1530
1530 GOSUB 1540:GOTO1560
1540 CLS:PRINT TAB(10)* P E R T
*"
1550 PRINT" (PROJECT EVALUATION
& REVIEW)":PRINT STRING$(32,131)
:RETURN
1560 PRINT:T=5
1570 PRINT TAB(T)"1 - ENTER NEW
PROJECT"
1580 PRINTTAB(T)"2 - LOAD OLD PR
OJECT"
1590 PRINTTAB(T)"3 - SAVE DATA"
1600 PRINTTAB(T)"4 - EVALUATE PR
OJECT"
1610 PRINTTAB(T)"5 - PRINT RESUL
TS"
1620 PRINTTAB(T)"6 - PRINT GANTT
CHART"
1630 PRINTTAB(T)"7 - END PROGRAM
1640 PRINT@448,STRING$(32,140);:
PRINT TAB(9)"YOUR CHOICE?";:SOUN
D100,1
1650 I$=INKEY$:IF I$=""THEN1650
1660 I=VAL(I$):IF I<0 OR I>7 THE
N SOUND100,1:GOTO1650
1670 ON I GOTO 1830,1890,2020,21
00,2280,2460
1680 GOSUB1540:PRINT" DO YOU WAN
T TO PREPARE A WORK- SHEET AT T
HIS TIME?"
1690 I$=INKEY$:IF I$=""THEN1690
1700 IF I$<>"Y"THEN CLS:END
1710 PRINT:INPUT" RESPONSIBILITY
";K$
1720 GOSUB 2650
1730 U$=" %
% % % ###
### ### "+STRING$(8,95)+
"+STRING$(8,95)
1740 PRINT#-2,CHR$(30);" PROJEC
T: "PT$:PRINT#-2," (WO
RKSHEET FOR RECORDING TIME ESTIM
ATES)";CHR$(13);CHR$(13);CHR$(28
)
1750 PRINT#-2," STEP
S RESPNS.
BEG END TIME FROM
TO"
1760 PRINT#-2," -----

```

```

-----
1770 FOR I = 1 TO N
1780 IF INSTR(R$(I),K$)=0THEN181
0
1790 PRINT#-2,USING U$;A$(I),R$(
I),A(I,1),A(I,2),E(I,1)
1800 IF INT(I/58)=I/58 THEN PRIN
T#-2,CHR$(12)
1810 NEXT I:PRINT#-2,CHR$(12)
1820 GOTO 1530
1830 IF PT$=""THEN 60
1840 GOSUB1540:PRINT"PROJECT CUR
RENTLY IN MEMORY:":PRINT PT$
1850 PRINT:PRINT"DO YOU WANT TO
ADD TO IT?"
1860 I$=INKEY$:IF I$=""THEN1860
1870 IF I$="Y" THEN GOSUB1540:GO
TO170
1880 N=0:GOTO150
1890 GOSUB1540
1900 LINEINPUT"FILE NAME: ";F$
1910 IF F$="?" THEN DIR:GOTO1900
1920 IF F$="" THEN 1530
1930 PRINT:PRINT" PROJECT BEING
LOADED:"
1940 OPEN"I",#1,F$
1950 X=0
1960 INPUT#1,PT$,Y:N=N+Y
1970 PRINT" ";PT$
1980 IF EOF(1)THEN 2010 ELSE X=X
+1
1990 INPUT#1,A$(X),R$(X),A(X,1),
A(X,2),S(X),F(X),E(X,1),E(X,2),C
(X)
2000 GOTO 1980
2010 CLOSE#1:GOTO1530
2020 GOSUB1540
2030 LINEINPUT"FILE NAME: ";F$
2040 OPEN"O",#1,F$
2050 WRITE#1, PT$,N
2060 FOR X=1 TO N
2070 WRITE#1,A$(X),R$(X),A(X,1),
A(X,2),S(X),F(X),E(X,1),E(X,2),C
(X)
2080 NEXT X
2090 CLOSE#1:GOTO1530
2100 GOSUB1540
2110 PRINT" DO YOU WANT TO CHAN
GE ANY OF THE DATA BEFORE EVA
LUATING THIS PROJECT?"
2120 I$=INKEY$:IF I$=""THEN2120
2130 IF I$="N" THEN 940
2140 IF I$<>"Y" THEN 2120
2150 PRINT:LINEINPUT" KEYWORD:
";K$
2160 U$="% %
###":FOR X=1TON
2170 IF INSTR(A$(X),K$)=0 THEN 2

```

```

250
2180 GOSUB1540
2190 PRINT A$(X):PRINT
2200 PRINT USING U$;" START NODE
",A(X,1):PRINT USING U$;" END NO
DE",A(X,2)
2210 PRINT USING U$;" EXPECTED D
URATION";E(X,1):PRINT USING U$;"
VARIANCE";E(X,2)
2220 PRINT:PRINT" CORRECT RECORD
?":PRINT @480," (PRESS <ENTE
R> TO END)";
2230 I$=INKEY$:IF I$=""THEN2230
2240 IF I$="Y" THEN I=X:FLAG=1:G
OSUB 550
2250 IF I$=CHR$(13) OR X=N THEN
GOSUB 1540:PRINT:PRINT" CLEARING
REGISTERS.....":FOR X=1TON:F(X
)=0:S(X)=0:NEXTX:PRINT:PRINT" RE
COMPUTING START/END TIMES...":GO
TO810
2260 NEXT X
2270 FLAG=0:GOTO1530
2280 GOSUB1540:PRINT:INPUT"RESPO
NSIBILITY";K$
2290 TIME=0
2300 GOSUB2650
2310 V=0:C=0:L=0
2320 PRINT#-2,CHR$(30);" PROJE
CT: "PT$:PRINT#-2
2330 PRINT#-2,CHR$(29);"

```

```

-----NODES----- EXP.
- START TIME- --END TIME-
- SLACK"
2340 PRINT#-2,"
STEPS RESP.
START END TIME EARLY
LATE EARLY LATE TIM
E "
2350 PRINT#-2,"
-----
-----
-----
-----

```

```

2360 U$=" %
% % % #####
##### ###.# ###.# ###
.# ###.# ###.# ###.# %
%
2370 FOR I=1 TO N
2380 IF INSTR(R$(I),K$)=0 THEN 2
430
2390 S1=F(A(I,2))-S(A(I,1))-E(I,
1):IF INT(S1)<=0 THEN C(I)=0 ELS
E C(I)=S1
2400 IF C(I)=0 THEN CR$=CHR$(93)
+"-- PAINTRITICAL" ELSE CR$=""
2410 PRINT#-2, USING U$;A$(I),R$
(I),A(I,1),A(I,2),E(I,1),S(A(I,1

```

```

)),F(A(I,2))-E(I,1),S(A(I,1))+E(
I,1),F(A(I,2)),ABS(S1),CR$
2420 TIME=TIME+E(I,1)
2430 NEXT I
2440 PRINT#-2:PRINT#-2:PRINT#-2,
" TOTAL EXPECTED TIME:";TIME
:PRINT#-2:PRINT#-2
2450 GOTO 1530
2460 GOSUB1540:PRINT:INPUT"RESPO
NSIBILITY";K$
2470 GOSUB2650:PRINT#-2,CHR$(30)
;CHR$(31);:PRINT#-2,TAB((40-LEN(
PT$))/2)PT$
2480 PRINT#-2,CHR$(30);TAB(25)" (
CRITICAL PATH SCHEDULE)"
2490 PRINT#-2:PRINT#-2,"RESPONSI
BILITY: ";:IF K$="" THEN PRINT#-
2,"ALL" ELSE PRINT#-2,K$
2500 PRINT#-2,CHR$(29);TAB(35)"
10 20 30
40 50 60
70 80 90
100"
2510 TT$="+-----+-----+
-----+-----+-----+
-----+-----+";PRINT#-2,TAB(35
)TT$
2520 FOR I=1TON
2530 IF INSTR(R$(I),K$)=0 THEN 2
630
2540 IF LEFT$(A$(I),1)="*" THEN2
630
2550 PRINT#-2,USING"%
%";A$(I);:P
RINT#-2,TAB(35)"!";:PRINT#-2, TA
B(S(A(I,1))+36);
2560 T1=INT(S(A(I,1))+.5):IF C(I
)=0 THEN T=INT(F(A(I,2))-E(I,1)-
.5):M$="X"
2570 FOR X= T1 TO T
2580 PRINT#-2,M$;
2590 NEXT X
2600 IF C(I)=0 THEN PRINT#-2:GOT
02630
2610 FOR X=S(A(I,1))+E(I,1)+1 TO
F(A(I,2))
2620 PRINT#-2,".":NEXTX:PRINT#-
2
2630 NEXT I:PRINT#-2,TAB(35);TT$
2640 GOTO 1530
2650 IF PEEK(65314)=4 OR PEEK(65
314)=6 THEN PRINT@480," DATA
BEING PRINTED NOW ";:RETURN
2660 PRINT@480," PRINTER IS
NOT READY";:SOUND100,5:FORX=1TO
500:NEXT:GOTO1530
2670 RETURN
2680 PMODE0,1:PCLEAR1:GOTO20

```

I thought I'd tell you a little about what it takes to get a BBS like CoCoLink underway.

Start in September, 1984, with a phone call to Patrick Simonis. Yes, there are several BBS programs available in the states, and he'll see which one might suit us. We are partial, but not in love with the idea of running under OS9.

In the meantime, Brian tells us that 'in theory', the construction of an auto-answer modem should not present any outstanding problems.

A hectic six weeks pass.

That's right, we had this idea that we might start a BBS - wonder what's happened to Patrick?

Patrick is duly phoned, and he has been active indeed. He has spoken to a number of suppliers, and has come up with a prog that runs under OS9, that will suit us 'fine! Where is it? Well 'they' said 'they' had sent it two weeks ago. Patrick will check.

What is Brian doing?

Preliminary work is complete and a mock up is about ready that looks as though it will work no sweat.

Two weeks later.

"Patrick, where is that program?"

"I've spoken to 'them' twice since your last call. 'They' hadn't sent it, but 'they' said they would. Second call reveals that it is still not sent!"

"Patrick - MAKE IT WORK!"

Two weeks later, the program arrives, and in the meantime, Brian's modem has worked. Work is underway on the boards, and the structure of the ROM. The cases are on the way and whilst not here, are the only part we don't have. Yes we should be a go for a December 1st start.

December 1st comes and goes, and still no cases - not that it matters, because we have tried to fire the bulletin board up with the new modem, only to discover that it doesn't work!

Brian, Kevin, Bob, Warren Warne and others go into a protracted huddle and decide that the ACIAs are the

problem. A fix is found, but not easily, and not without a lot of late nights, and Brian gets the thing to fire up in Brisbane. We look set for the 10th December!

Brian takes a day off work, and comes down to 'set up' the BBS.

It might have worked in Brisbane, but it doesn't work on the coast! Brian goes home without joy!

It is getting on for Christmas, I'm about to leave, and Brian is due back on 22nd. In my absence, Brian and Kevin fire the thing up for the first time. Gordon Bentzen logs in as our fist user (and, I might add, we never heard from him again!). But all is not well, and the CoCo being used starts to heat up, so the BBS is switched off till I return on 7th Jan.

Kevin fires it up that morning and Ivor Davies is in almost immediately! In fact the thing goes pretty well for about a week, before we discover that people hanging up prior to logging off are also hanging up the system.

This is a problem that is yet to be solved, but since January, we've begun to become proficient in the manipulation of data around the board. We also have made a start on the input of some data into the thing, and the program has been changed.

The week after that happened, we were pretty busy trying to activate the various modules and trying to discover why it was hanging up more often. Kevin did sterling work and brought most of that under control in a very short space of time.

As I write this, Bob T has just been on the phone to let me know that the 'Visitor' module is ready, and is being sent in the morning.

On the hardware front, the 64K grey case being used, sports a RAINBOW BITS multi pack interface with a RAINBOW BITS modem and disc controller attached. The drives are 40/80 track Nationals. The system currently boots OS9 on 40 tracks, and is then switched to 80 tracks.

When you consider that all the folk involved in this project have day, or night jobs, and are heavily committed to the production of the magazines and tapes as well, you can see that it has been no mean task!

STOP PRESS

GAMES COMPETITION.

Entries to the Games Competition close on 31st May. Already some excellent ORIGINAL work has been turned in, so your work will have to be good!

Games can be adventure or graphic, and all submissions since August last year will qualify for consideration.

Additionally, if you have a particular preference for a program, and would like to see it win a prize, then let us know!

The prizes are:

1. a disk drive from Software Spectrum.
2. a 128 K upgrade from Blaxland Computer Services.
3. a Tandy Speech - Sound pack from Bayne and Trembath.

We thank the suppliers for their involvement, and want to point out that the prizes have been donated with a view to encouraging you to work harder at your programming!

PRICING.

Last issue, I said that we were looking at raising the price of Rainbow again. We were very tempted to do that this month, but decided to wait and see what happens to the Australian Dollar for another month.

I believe that the dollar will either remain stagnant, or drop in value - in which case I should be putting the price up now. But I have my heart set against raising the price if at all possible.

None the less, if the Aussie dollar does rise, or remains stagnant this month, the price of Rainbow will have to rise, and I'm sorry but I can't help it!

If you are the betting type, and you want to update your subscription, even if your sub does not expire this month, I won't stop you.

Graham.

A Guide To The Do's And Don'ts Of Modeming

By Wayne Day

By the time you read this, likely all of the Christmas presents will have begun to get a lot of usage, especially if a modem or a new terminal program had been hung in the stocking!

Quite a few new users will have shown up on CompuServe's Color SIG (Special Interest Group), where I serve as the systems operator (SYSOP), and they will begin to learn about the "new" world of telecommunications.

So now might be a good time to review some of the "do's" and "don'ts" of modeming.

First off, get familiar with your equipment and software setup. Knowing your hardware and software capabilities, and limitations, will save you valuable connect time, which is even more important if you're calling long distance or connected to one of the commercial information services.

Read the documentation that comes with the software, paying special attention to instructions on how to capture data (if your software supports that function).

For example, some programs require you to open a buffer manually in order to save anything you receive online, while others allow the host system (the computer you're calling) to open and close your receive buffer automatically. This latter method is known as "capture characters" since the host sends a particular character to your software, telling it to start "capturing" the data that follows.

On other terminal programs, however, you need to defeat word-wrap and high resolution character screens prior to capturing data in the buffer. Again, the

point is to read those instructions very carefully.

Knowing the operating hours of those BBSs is important, since it might save you a long distance phone call if the system is unavailable. In November, we listed a group of 92 known BBS systems that were of special interest to the CoCo user, and this month we bring you an update, bringing the total up to 115 operating BBS systems.

Other "Do's" to remember:

If you're calling a single-user BBS, be considerate of other users and don't dawdle on the system unnecessarily. Others may be waiting to call (and, you might wind up having to wait sometime).

Make note of any message content policies that the SYSOP may establish, and abide by his wishes. Some BBSs restrict commercial messages and others prohibit them entirely. Ditto on "adult" language.

On the other hand, there are the "don'ts:"

Don't try to "crash" the system, just to see if it can be done; robbing others of the chance to use a BBS is pretty childish. Since most BBS systems make a note in the user log each time someone logs on, you may find that you're no longer welcome on the board next time.

Don't ask folks to help you out with something, and expect them to be mind readers. Asking a question like "I wrote a program and it doesn't work. Why?" doesn't give anyone enough information to work with. The more information

you can put in a question, the better your chances for a good answer!

Don't forget to read all of the messages on a BBS system. Quite often you'll stumble across some piece of obscure information that will really save the day — two weeks later!

And finally, don't forget to have plenty of paper and a pencil nearby to take notes on other BBS systems that you might be interested in, as well as a place to jot down that neat little poke everyone's been talking about!

Whew!

Letters, Letters, Letters

Q: Is it possible to allow another CoCo user to call up my modem, then let him store programs on my disk drives, and use my printer? The reason I ask is that many students at the university I am attending could benefit from a service such as this, while I could make some spare money.

A: Sure, it's possible ... that's basically what services such as The Source and CompuServe do — allow you to use *their* computer and disks.

On a CoCo, you'd need an auto-answer modem (to answer your phone automatically and connect it to the computer), as well as a remote terminal driver, such as *REMOTERM* (sold by Star-Kits) or Dan Downard's *REMOTE.BIN*, published in *THE RAINBOW* in November 1984.

The remote terminal driver basically lets the person calling in on the modem act as if he were sitting at the keyboard of your computer; all of the data he types on his terminal is the input to your computer, and all of the output

from your computer goes to his terminal.

There are some limitations, though. Graphics screens cannot be displayed in this manner, since you will be limited to using the normal ASCII character set (*CHR\$ 0-127*). Additionally, any programs that use *PRINT@* statements must be converted to normal *PRINT* statements, since there is no way to control the other user's cursor or screen position. That also means that a *CLS* won't work, either.

If you're setting up something like this with a friend, there's no problem since you know who it is that has control over your computer. Remember, though, that since he acts like he's sitting at your keyboard, he can even do a *DSKINI* and wipe out all your disks!

Therefore, if you're going to be doing this with more than one person, or providing the service commercially, consider the security requirements you'll have to make:

For example, user 'A' should not have access to user 'B's' files, to protect confidentiality.

No user should be able to harm the system itself. That means you'll have to provide some way to protect against a *DSKINI* or *POKE* into the wrong place, and that usually means a remote time-sharing monitor, such as the *TSMON*, part of the Radio Shack OS-9 Operating System.

Q: Your column ("CommLink", October 1984) made the first mention I've ever seen of using a party line for data transmissions. How disastrous is it if someone else on the line picks up the phone? Can they tell something is going on, or will they try to call a repairman to fix the funny noises?

A: There's a two-fold answer to this question: a practical one, and a legal one. First, the legal considerations.

Consider this scenario: It's 2 a.m., and you've finally gotten in to that popular BBS on the West Coast. You've got a lot of downloading to do, and waiting for your chance on the BBS has been a drag.

About halfway through your first download, your neighbor, the one on your party line, is awakened by the smoke detector in his house -- he has a fire, and obviously needs to call the fire department, but can't. It seems there are some funny noises on the phone line.

Far-fetched? Possibly, but not too far-fetched.

The simple fact is that all states have laws that require you to hang up the receiver of a party line immediately
April, 1985

when told the line is needed for an emergency call to a fire department, or police department, or for medical aid. In Texas, it's an offense that brings a fine of not more than \$500 or imprisonment for a term not exceeding one month, or both (Southwestern Bell Telephone Company Phone Book, 1984).

If you're online, the chances are you won't hear someone on a party line picking up the telephone.

Direct-connect modems must be registered with the Federal Communications Commission, and the F.C.C. restricts direct-connect modems to the extent that they may not be connected to a party line or pay telephone.

Thus, two big reasons why a party line can be hazardous to your modeming health.

Let's consider a one-party line, where someone picks up an extension telephone in some other part of the house.

Depending on what you are doing (downloading using an error-correction protocol like XMODEM or just sending and receiving straight ASCII data), someone picking up one of the other phones in your house can have from a slight to a disastrous effect on your telecommunications session.

From personal experience, there have been times when someone else picked up a phone in my house, and I never knew it (our neighbor still wonders about those funny tones on our line!).

During testing of a new terminal program that uses the XMODEM protocol, we have picked up the phone and shouted into it, played music into it, and generally banged it around, with no ill effects other than causing an error in the transmission of the current packet of information, which was re-sent and received OK after we quit trying to goof it up.

Other times, though, before the installation of the computer's own phone at my house, I've been bumped off by someone picking up the phone -- blew me completely off the system I was talking to.

So, it's an iffy situation; sometimes it won't harm anything, and other times it's bad news.

The ultimate solution might be to have a second line installed if modem use and more normal telephone usage conflicts crop up. It's been a lifesaver (mine) at my house!

Two Questions, One Basic Answer

Q: I called one of the numbers you listed in the BBS Roundup (*The Rainbow*, November 1984) and got a

message saying that the number had been disconnected. Can't you keep up with the numbers you list?

Q: How come you didn't list any of the Coco BBSs in my area?

A: Let's kill two birds with one stone here, and explain a bit about how I go about collecting the BBS numbers published with this column.

Since no one can be everywhere at one time, it's only possible to list the BBS phone numbers that we become aware of, usually through messages on the Color SIG on CompuServe, on other BBSs, and in letters to THE RAINBOW.

Likewise, if a BBS goes out of business, we usually don't hear about it until someone writes in, or leaves us a message saying "you blew it!"

Therefore, we need your help in making the BBS listings in THE RAINBOW the most accurate and up-to-date that we can. If there's a CoCo BBS in your area, let us know about it. If one goes off-line, you could also drop us a line to let us know.

Every couple of months, we'll try and publish a list of revisions such as the one included at the end of this month's column to keep you up to date. Additionally, I maintain a current list in the SIG's database on CompuServe, so CompuServe subscribers can get the whole shooting match at one time. That list is usually updated at least monthly.

Before I put a BBS on the list, I call it to make sure that, indeed, there is a BBS there, that our information is correct, and that the BBS is open to the public.

Some Random Thoughts

John Lovell, the SYSOP of The CIC BBS in Miami, has a series of modifications to the *Bee Color BBS* program, and he's making arrangements to make those changes available to SYSOPs who are already running the *BEE* program. Give his BBS a call (305 751-6809) if you're interested in knowing more.

If you're a new user to CompuServe, you might find yourself baffled by the hundreds of things you can do on the system. Well, there's a book written just for you, *How to Get The Most Out of CompuServe* (Bantam Books, New York, \$12.95). The authors, Charles Bowen and Dave Peyton, are SYSOPs on two of CompuServe's Special Interest Groups, and have compiled the most comprehensive sourcebook for CompuServe users yet. It's available at most book stores, or directly through CompuServe (Go PCS-54).

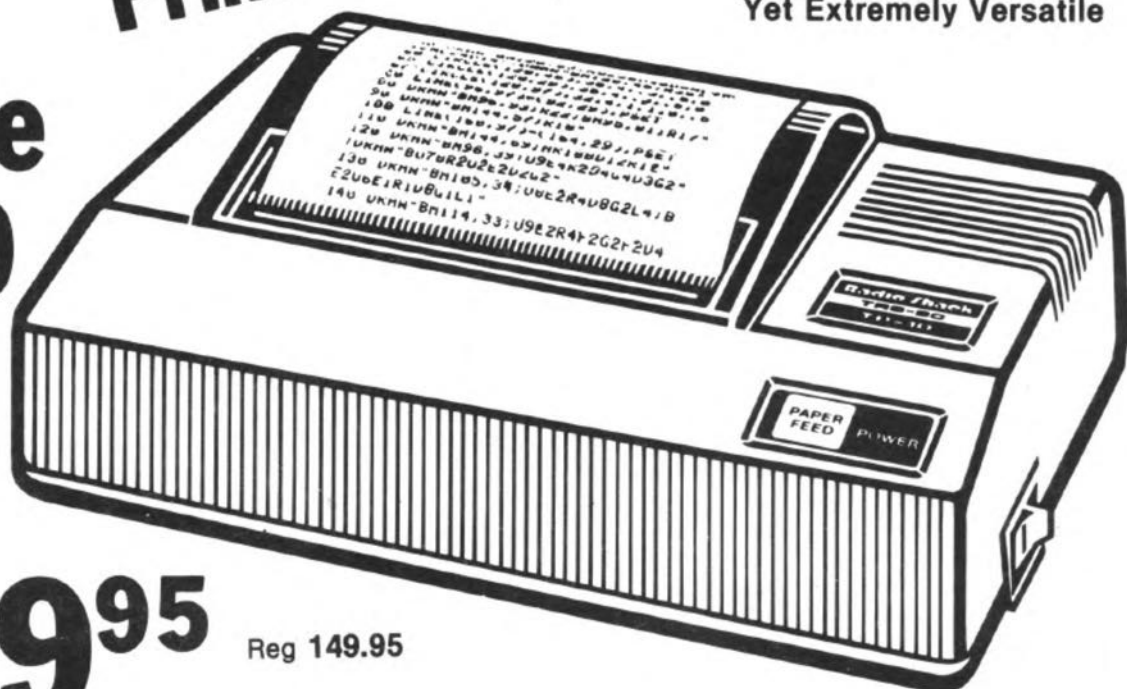
Tandy

ELECTRONICS

Low-Cost Thermal Matrix Printer with Graphics

Amazingly Compact,
Yet Extremely Versatile

Save
\$20



129⁹⁵

Reg 149.95

TP-10 Thermal Printer. Pay less for the perfect compliment to your CoCo or MiCo! Whisper quiet operation prints 32 characters per line at 30 characters per second on 10.47cm wide thermal paper. With elongation mode for expanded print. Special repeat

function makes graphics programming easier! CoCo compatible serial interface only (600 baud). 94 ASCII, 16 graphics character set. 10 or 5 cpi pitch with 1/6" line spacing. Measures 7.62 x 20.32 x 12.7cm, weighs 1.47kg. 240VAC, 50 Hz, 18W. 26-1261

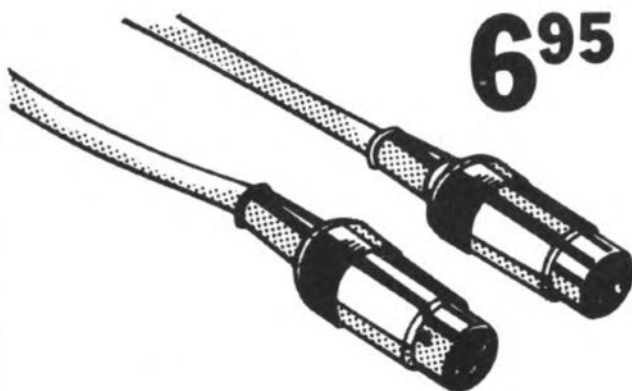
Thermal Paper



5⁹⁵

Thermal Paper. Packet of 2 thermal paper rolls for your TP-10. 26-1332

MC-10 Interface Cables



6⁹⁵

4-Pin to 4-Pin serial cable to connect your TP-10 to your Color Computer. 26-3020

SALE! Exciting Color Computer Games

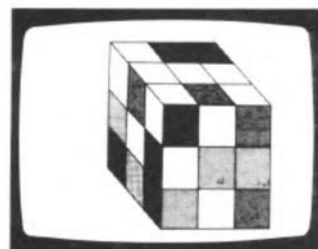
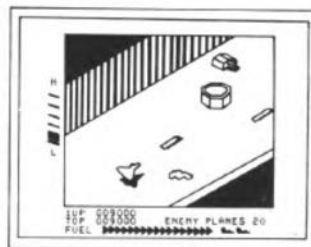
Save \$20! ZAXXON™ Fly your jet through defenses, force fields and outer space — destroying enemy ground installations, fighting planes and dodging missiles all the way! Survive those dangers to confront the deadly Zaxxon robot himself! Requires 32K, joysticks and tape recorder. 26-3062 Reg 59.95 **Now 39.95**

Save \$20! Micro Painter. Turns your CoCo into an electronic colouring book! Choose from three palettes, each of four colours. Save your more artistic results on optional recorder. Comes with 8 pictures and more available. 26-3077 Reg 49.95 **Now 29.95**

Save \$10! Dungeons of Daggorath™ Defeat the evil Wizzard by descending through his dungeon, battling a succession of beasts to gain better weapons and greater strength. The further you progress the more fearsome the beasts, until the ultimate confrontation — with the Wizzard himself! 26-3093 Reg 49.95 **Now 39.95**

Save \$20! Colour Cubes. Use your CoCo to crack *that* crazy Rubik's Cube! Recall up to 255 moves. Use an optional cassette recorder to save your position. 26-3075 Reg 39.95 **Now 19.95**

Save \$10! Reactoids. Prevent meltdown at the atomic reactor! Fast, fearless use of the reactoids can contain and control the atoms! Can you save the world? 26-3092 Reg 34.95 **Now 24.95**



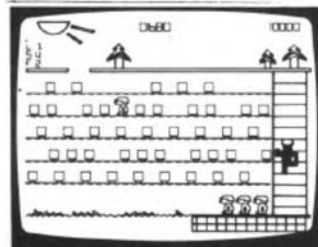
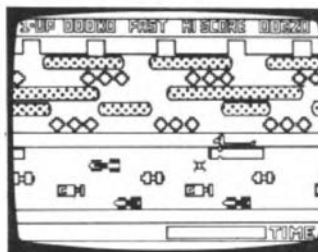
NEW! Exciting Games Software!

Frogger. Safely manoeuvre your frogs across the busy highway and swiftly flowing river to their homes. Look out for speeding traffic, sinking turtles, hungry snakes and crocodiles! 26-7304 . **29.95**

Cuthbert in the Mines. Help Cuthbert dodge the mining carts, climbing to the surface and safety. But beware the fireball shooting Demon! Brilliant graphics and sound effects. 26-7306 . **29.95**

8 Ball. Play pool from your chair! Line up the shot with your electronic cue, select how much power you'll use — you can even put spin on the cue ball! Ideal for solo play too. 26-7311 **29.95**

Scramble. The ultimate mission! Pilot your way through 4 levels of enemy defences, including UFOs and blazing meteorites! Then through a maze for one chance of bombing the target. 26-7314 **29.95**



WE SERVICE WHAT WE SELL!

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*

Tandy ELECTRONICS

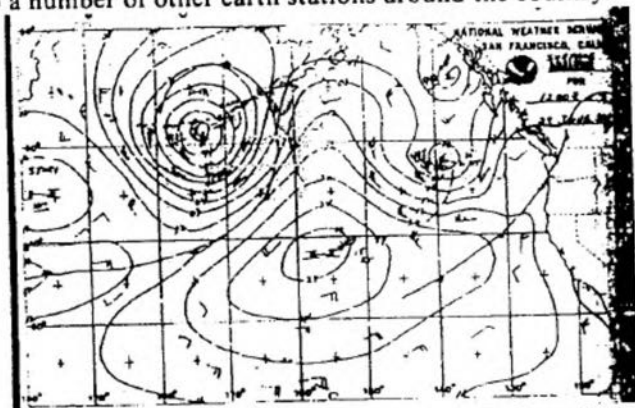
A DIVISION OF TANDY
AUSTRALIA LIMITED
INC. IN N.S.W.

Nearly
350 Stores
Australia-
Wide

Weather . . . Or Not?

By Martin H. Goodman, M.D.

Hovering roughly 22,300 miles above the equator is a remarkable device. Since 1974, this Geostationary Operational Environmental Satellite (GOES) and its predecessors have been sending pictures of the earth, taken in visible and in infrared light, down to earth stations via a microwave link. Every day, a dozen or more images of the cloud cover over our part of the planet are processed by a government ground station, the images enhanced, and state and national boundaries added. These enhanced images are then "uploaded" back to the satellite by microwave. The satellite retransmits the enhanced images to a number of other earth stations around the country.



One such secondary station is near me in Redwood City, Calif. There, meteorologists examine these precious photographs, and are able to draw two synoptic charts per day of weather fronts and such. (A "synoptic chart" is that map with all the wavy lines indicating areas of equal temperature, warm and cold fronts, and sometimes wind and sea information as well.)



The enhanced satellite photos and the hand-drawn synoptic charts are sent in facsimile (fax) format (a 40-year-old electronic protocol for picture transmission used by wire services as well) over a telephone line to a transmitter. In my case, that transmitter is located at Point Reyes, Calif. This fax signal is then rebroadcast, primarily for ships at sea, on a number of high frequency shortwave bands.

Twice a day in my area, a series of visible and infrared photos of my part of the world taken from space, along with superb meteorological charts, are sent out over shortwave frequencies. Until recently, it took, in addition to a shortwave receiver, \$5,000 worth of mechanical facsimile machines to turn that warbling fax signal back into a picture.

About The Authors

As you might well imagine, my good friends, the authors of *Graphicom*, have an abiding interest in graphics data. They decided to use the processing capabilities of the CoCo and its amazing built-in analog signal processing capabilities to turn a fax signal back into a picture and turn that picture into hard copy.

When they were done, they had a tiny assembly language program (only about 2K long) that uses the zero crossing detector of the cassette port of the CoCo to receive a fax signal from any BFO-equipped shortwave receiver and process it, turning that signal into a 54K-byte picture inside the CoCo.

Now, a 54K image is exactly nine times more information than the CoCo can display on the screen at one time, so the program they wrote, *WEFAX*, has built-in routines to allow the user to pan the CoCo Hi-Res screen over that larger "virtual" image.

They also supplied routines for saving the pictures to *Graphicom* format picture disks (two pictures to a disk, 54K is a lot of data!), and for transferring those pictures from the computer's memory to paper using an Epson dot-matrix printer. It will not surprise users of *Graphicom* to learn that all program control is via a joystick and one fire button.

Technical Details

Due to the 64K RAM memory limitation of the CoCo, my friends were only able to process roughly one-half of the vertical and horizontal data contained in the fax transmission. Even so, the images produced are of

remarkable quality. (See the sample pictures included with this article.) Indeed, when they took the pictures down to the Weather Service office in Redwood City, it turned out the CoCo WEFAX hard copy was equal, if not superior, in quality to the hard copy produced by some of the \$5,000 mechanical fax machines there.

Rather than totally throw away every other line of received fax data, the WEFAX program does a remarkable thing. During reception of the picture, it uses every other line to construct on the CoCo Hi-Res screen a 6K compressed, representative image of what it's receiving. When reception is complete, you are then switched to the 54K image and your pan function.

As you will see in the official WEFAX documentation that follows, a minimal amount of hardware is needed to interface a shortwave receiver to the CoCo cassette port. This simple circuit, consisting of two LEDs (used as zener voltage limiting diodes), a single resistor, and a single capacitor, is used solely as a voltage limiter and DC filter. All other processing of the signal is done by the CoCo's internal hardware and the magic of the WEFAX software.

Required Hardware

A 64K, one disk drive equipped Color Computer (any model) and one joystick are required. An Epson dot-matrix printer is needed to produce hard copy. Almost any general communications shortwave receiver with a BFO (and preferably LSB and USB selection capability) will do to receive the pictures. Your receiver does need a minimum of frequency stability in order to receive a clear WEFAX picture, but most receivers costing more than \$150 will suffice.

What Frequencies To Try

There are hundreds of stations all over the world broadcasting weather map data. WEFAX was written explicitly for weather map reception, and while it does receive other fax data (like UPI wirephoto information), images produced are likely to be somewhat blurry and/or distorted. To get you started, here are some of the more useful frequencies to listen to to find WEFAX data. All frequencies are in kilohertz.

East Coast, Station NAM in Norfolk, VA. 3357 kHz, 4975 kHz, 8080 kHz, 10865 kHz, 16410 kHz, 20015 kHz

West Coast, Station NMC in Point Reyes, Calif. 4346 kHz, 8682 kHz, 12730 kHz, 17151 kHz

West Coast, Station WWD in La Jolla, Calif. 8646 kHz, 17411 kHz

The Program And Its Source Code

The documentation follows the BASIC listing. Carefully type this program in *exactly* as it appears. Be sure *not* to alter even the line numbers. The program has a built-in checker for errors, which, when you run it, will tell you if you made an error typing in the data statements, and the line number where the first error appeared. Those of you receiving RAINBOW ON TAPE will, of course, be spared this agony. When you have the program typed in correctly, typing RUN will make it work.

For those who do not receive RAINBOW ON TAPE, the program WEFAX/BAS is available for downloading from CompuServe's Color Computer Special Interest Group (SIG) in database XA2. In addition, also posted in XA2 is WEFAX.ASM, the full assembly language source code (written with a version of the Micro Works SDS Macro

80C). This will be of interest and value to those who wish to insert their own printer routines for different printers. These people should carefully study the existing routine. It prints successive rows of dots in overlapping fashion. To produce a reasonable picture, the printer you support must be capable of doing this, too. For those without this editor-assembler system, note the printer routine is a separate module at the end of the program, allowing you to easily substitute your own routine even if you don't have the source code.

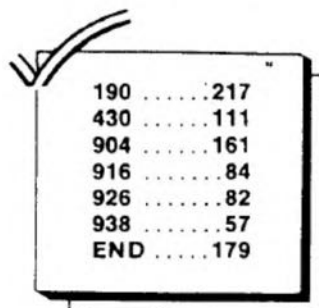
For those who cannot download the source code from CompuServe, I will provide you with a copy if you send me a disk, a mailer, return postage and \$10 for handling. Address requests to Marty Goodman, 1633 Bayo Vista Ave., San Pablo, CA 94806.

I will also provide you with an experimental version of the program that not only receives, but is capable of sending a fax image as well. However, I must make it clear that I will *not* support either program, or answer *any* questions about it, because I'm not the original author, and neither I nor the authors have time to do so. Your comments are, of course, welcome, but it is very unlikely we will be able to respond to questions.

Both the authors and I would be delighted to receive any enhancements you may write for the program, especially alternative printer routines. I'd like to see a routine for the C. Itoh Prowriter printer. I haven't had time to write one yet for myself. If you send us such a routine, you'll be on our "list" for receiving other goodies.

The Documentation

I'll let the authors themselves provide you with further instruction on using their WEFAX program. Following this listing is material mainly consisting of *Graphicom* screens the authors meticulously and laboriously drew, just in order to better share this program with you.



190	217
430	111
904	161
916	84
926	82
938	57
END	179

The listing:

```
10 REM THIS PROGRAM IS PUBLIC DO
MAIN
20 PCLEAR4
30 F1=651:REM 60 LPM ADJUST
40 F2=901:REM 90 LPM ADJUST
50 F3=1027:REM 120 LPM ADJUST
55 POKE150,1:REM PRINT=9600 BAUD
60 CLS
70 PRINT" WEATHER FACSIMILE RE
CEIVER"
80 PRINTSTRING$(32,"-");
90 PRINT
100 PRINT"THIS PROGRAM WILL RECE
IVE FM FAX";
110 PRINT"AT 60, 90 & 120 LINES/
MINUTE."
```

```

120 PRINT"CONNECT THE CASSETTE E
AR (BLACK)";
130 PRINT"PLUG TO THE RADIO'S EA
R JACK OR"
140 PRINT"CONNECT IT ACROSS THE
SPEAKER."
150 PRINT"CHECK THAT THE RADIO I
S TUNED IN";
160 PRINT"AND THAT THE BFO IS TU
RNE D ON."
170 PRINT"AFTER THE PROGRAM IS R
EADY YOU"
180 PRINT"CAN USE THE TUNING MET
ER TO"
190 PRINT"ADJUST THE RADIO."
200 PRINT
210 PRINT"PLEASE WAIT..."
220 AD=&H600:LI=900
230 READA$,CS
240 IF A$="X" THEN 320
250 FOR I=1 TO 64 STEP 2
260 A=VAL("&H"+MID$(A$,I,2))
270 POKE AD,A:CS=CS-A:AD=AD+1
280 NEXT
290 IF CS THEN PRINT"DATA ERROR
IN LINE";LI:END
300 PRINT@462,944-LI
310 LI=LI+1:GOTO 230
320 A=INT(F1/256):POKE&H600,A:PO
KE&H601,F1-256*A
330 A=INT(F2/256):POKE&H608,A:PO
KE&H609,F2-256*A
340 A=INT(F3/256):POKE&H610,A:PO
KE&H611,F3-256*A
360 PRINT"RECEIVE *60* *90* *120
* LPM"
370 PRINT
380 PRINT" *PRINT TO EPSON PR
INTER*"
390 PRINT
400 PRINT" *REVERSE VIDEO THE P
ICTURE*"
410 PRINT
420 A$=CHR$(PEEK(&HC000))+CHR$(P
EEK(&HC001))
430 IF A$<>"DK" THEN PRINT:PRINT
:PRINT:GOTO470
440 PRINT"DISK *VIEW* #1 ***
#2 ***"
450 PRINT"PICTURE *LOAD* ***
***"
460 PRINT"STORAGE *SAVE* ***
***"
470 PRINT
480 PRINT"TUNING I-----+I"
490 PRINT"METER: 1500 2300"
500 PRINT
510 PRINT"JOYSTICK RESTART
SCAN"
520 PRINT"USE WHEN PHASE HOL

```

```

D PHASE"
530 PRINT"RECEIVING: <-- SCA
N -->";
540 EXEC CS
900 DATA0285050A004D00E703820704
0033009904000800002600720F718634
B7FF030F,1740
901 DATA660F671700E3260EB7FFDEBE
C0066F846F01AD9FC004B7FFDF7FFF40
7F098610,3846
902 DATA03D78E0400A6848A40A780
8C060025F51703338D0220DE8D57080B
0002840D,2661
903 DATA100002821216000281041B22
048F021C440332090E660011090E7700
1D090E88,1154
904 DATA001C80800000788D28139568
03D61A9C6803D5808000006786022002
86039761,2490
905 DATA8D0F131568034B1A1C68034A
808000004E17027235109F628D2ADD64
8D773410,2244
906 DATA1700908D1F10936427073510
17008420E71700EE8D4D27EBAEE43003
EC84AD8B,3021
907 DATA3510206F8D18FC015A445454
8D1B2607C1082302C60839C10423FBC6
0439B7FF,2850
908 DATADEAD9FA00AB7FFDF39B7FFDE
BEC000B7FFDF8C444B39B6FF00B40139
8DF827FC,4792
909 DATA9766398DF19166270B8E222E
301F26FC8DE49766399E628D6B916422
04D16424,3391
910 DATA073005E6842AF0398D619165
22F3D16525EF39CE04008D536D012A04
96651F89,3174
911 DATA3402E0E45CE7E4C6203D33CB
8D366D012A0496641F893406E6842A10
33C6E6C4,3524
912 DATAA68D001FA7C4E78D00192014
E6C6C840E7C64CA16123F533C820A6E4
6A6226EC,4040
913 DATA326339CFEC84841FC41F39A6
0244444444E602C40F39B6FF205A2705
B1FF2027,3274
914 DATAF839C6208DF02708C6408DEA
27028DE654D16727158E05409667D767
E686CA40,3880
915 DATAE7869667E686C4BFE7863986
34B7FF03863CB7FF0186D6B7FF208680
B4FF0027,4605
916 DATA053DAC842008DC52DD56DC50
DD54862AB7FF208680B4FF0027043D12
200ACC26,3373
917 DATA02DD56CC0E00DD548634B7FF
019E4086D6B7FF208680B4FF0027038E
0001862A,3555
918 DATAB7FF208680B4FF0026029E42
301F26FCA69439DF483A3386DF4C8601

```

A7844F5F, 3616
 919 DATADD4AB6FF205CC1262504A101
 2005B1FF2027F2D74E1DA18BD34ADD4A
 9348251B, 3557
 920 DATADD4AD64EF7FF02D14F698424
 0430012005CC0001E70112C606200F86
 01B4FF00, 3018
 921 DATA260617FE7716FD8EC6049C4C
 25B4DC48934A2F07A18B83000126F917
 FF2D399E, 3332
 922 DATA52DE448660208C5F9E50DE46
 862020838DF55F8DEA8DF0C6608DE48D
 EAC6C020, 4425
 923 DATADE5F2006C6082002C610308D
 FD123ACE00408608E680E7C04A26F98D
 49861197, 3397
 924 DATA4F8E26029F52CE0E00DF5033
 C820DF54308901209F568DB8DE54DF50
 9E569F52, 3497
 925 DATA8CFE0225E6397FFF22CEFFC6
 A75AA75CA75E860444C6074424063341
 A7C02002, 3602
 926 DATAA7C15A26F23986F0B7FF22CE
 FFC6A75AA75DA75F860E20DCB690F601
 5BC13F26, 4381
 927 DATA015C3D584958498E2602308B
 B6015A813F26014C3086203800000000
 00000000, 1695
 928 DATA00000000000000008DBD8DCDFC
 015A340617FD6BFC015A10A3E12702BD
 BB17FD83, 2983
 929 DATA27E9398DA18E260263808CFE
 0225F939108E0E0C620A680A7A05A26
 F9308840, 3427
 930 DATA108C260025EE39108E0E00C6
 20A6A0A7805A26F9308840108C260025
 EE392602, 2740
 931 DATA262226426E026E226E42B602
 B622B642C6002002C611D760D661C103
 26028D24, 2738
 932 DATA17FF43318CD88609AEA13422
 9661810326068D9C8D7620048D728DAB
 0C603522, 3091
 933 DATA4A26E539B7FFDEFEC006CC02
 00EDC4CC1102ED428E0E00AF44AD9FC0
 04B7FFDF, 4263
 934 DATAD660260FC600A6854C1026FB
 E95CC11723F439C62BA6854C1026FBDA
 5CC14323, 3644
 935 DATAF439C6ED2002C6F7340417FE
 D9D665C00686033DDB64EBE0D7608602
 97618D0C, 4102
 936 DATA7FFF407F098617FC8D26FB39
 3476B7FFDEBEC006D661E7846F01CC0E
 00ED04CE, 4147
 937 DATAFFFF9660C6183D1083013325
 03C30002334183001224F9C30013E703
 1F30E702, 2785
 938 DATA108E0018AD9FC004LD1F5C04
 EC02108311012602CB025CC1122303C6

014CED02, 2497
 939 DATA313F26E0B7FFDF35F617FC2A
 260C7FFF407F098617FC2516FB3C3917
 FE48B6FF, 3905
 940 DATA228401263886FE976F8E2602
 861B8D2386338D1F86178D1B8D24861B
 8D158633, 2797
 941 DATA8D1186018D0D8D1630890540
 BCFE0225DB860DB7FFDEAD9FA002B7FF
 DF398DA9, 3845
 942 DATA860D8DEF861B8DEB864C8DE7
 86008DE386038DDF866034128601975F
 A684108E, 3786
 943 DATA0058C6084969A05A26FA3089
 00C024EC8E0058C608A680438DB95A26
 F8351230, 3277
 944 DATA014A26D23900000000000000
 00000000000000000000000000000000
 00000000, 380
 945 DATA, 1560

WEFAX Official Documentation

Have you ever wondered what the weather is doing far out at sea, but lacked the expensive equipment necessary to receive the charts?

Here is a low cost way you can. Just get a Radio Shack 64K Extended BASIC Color Computer, joystick and cassette recorder with cable.

The one other thing you need is a good shortwave receiver that can receive single sideband (SSB) transmissions. SSB is used because it's much more efficient than the regular AM and FM modes used by local broadcasters, but it does require a receiver that has a beat frequency oscillator (BFO) to make any sense of the signal. Many of the newer digital-readout portable shortwave radios (and some cheaper sets) have this, but you probably won't find it on a typical multiband radio or a "jam box." If your set has this capability, there will be a switch to turn the BFO on or off (on better receivers this will have separate CW and SSB positions). Follow the instructions you got with the radio for tuning in SSB transmissions.

WEFAX is written in machine code to get the required operating speed. The machine code for WEFAX is loaded into memory by a BASIC program. The data that makes up the second half of the listing is the actual machine code, along with an error checking number for each line.

Type in the program exactly as it is printed. Be careful to get all the spaces in the text of the menu. When typing is complete, save it two times on tape (or disk) and then type RUN. If you get an error, then reload the program and correct the error. Now save it, then RUN it again. When it loads properly you should see the menu page appear.

WEFAX RECEIVE MENU PAGE

RECEIVE ♦60♦ ♦90♦ ♦120♦ LPM

♦PRINT TO EPSON PRINTER♦

♦REVERSE VIDED THE PICTURE♦

DISK ♦VIEW♦ #1 ♦♦♦ #2 ♦♦♦
 PICTURE ♦LOAD♦ ♦♦♦ ♦♦♦
 STORAGE ♦SAVE♦ ♦♦♦ ♦♦♦

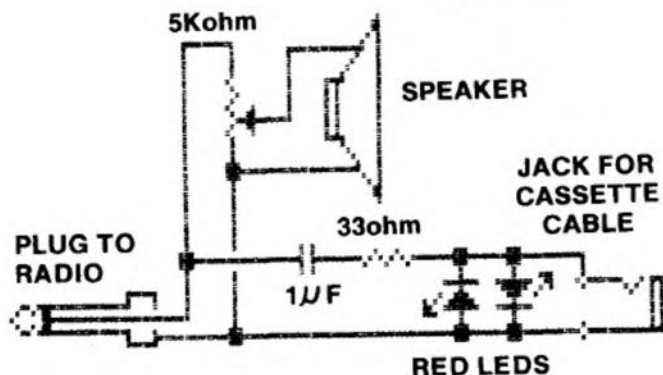
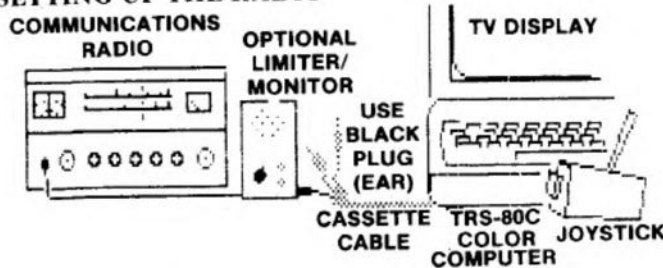
TUNING I-----I
 METER: 1500 2300

JOYSTICK RESTART SCAN
 USE WHEN PHASE HOLD PHASE
 RECEIVING: <-- SCAN -->

Setting Up The Radio

If you find that the computer interferes with the radio, try moving the two apart. Another thing you can try is coiling the cassette cable around a ferrite rod. It also helps to use shielded (coax) cable to an antenna 20 or more feet away.

SETTING UP THE RADIO



OPTIONAL LIMITER/MONITOR

The optional limiter/monitor shown here helps reception of signals with fading and noise. It also lets you monitor the signal at a comfortable (and adjustable) level while maintaining a good signal into the computer.

The Menu

The menu screen of *WEX* receive is in ways like the control panel of a mechanical facsimile receiver. Try moving the joystick around; you will see that various functions will be highlighted.

To select any function, move the joystick until the function you want is highlighted and tap the button to activate it. When that function is completed or if you don't select any function and tap the button you will pan over the picture (see PAN).

To abort any function, hold down the button until it stops (this may take a second or two for some of the functions like *PRINT*). Don't tap the button, just hold it down until the operation stops.

Tuning In A Fax Station

Here are two stations that broadcast continuously:

West Coast/Pacific: NPM (Hawaii) 14.823 MHz Lower Sideband

East Coast/Atlantic: NAM (Virginia) 8.027 MHz Upper Sideband

Turn on the radio's BFO (switch may say CW or SSB). Tune in the station and adjust for highest reading on the S meter (on some radios you may have to turn the BFO off temporarily to get a proper reading). Notice the jiggling pointer (black rectangle) on the tuning meter. Adjust the BFO control until the movement of the pointer is mostly contained between the 1500 and the 2300 Hz marks. Weather charts are mostly white so the pointer will spend more time on the 2300 Hz mark when tuned to the proper sideband.

Other signals: I#-----I
 1500 2300

Phasing interval: mostly black. I#-----I

Satellite picture: varying I#-----I

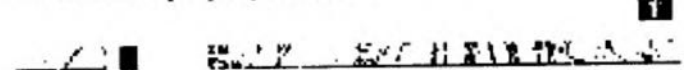
Effect of noise: random I#-----I

Signal + noise: I#-----I

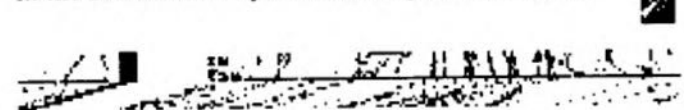
Receiving The Picture

Now select 120 lines per minute (for most stations), and tap the button, move the joystick to center bottom and you are now receiving a picture.

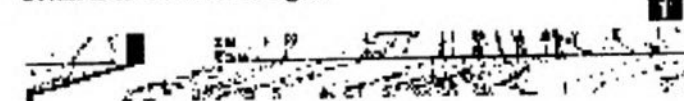
If you start when a chart is in progress, you may see the chart not properly centered.



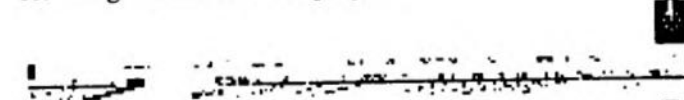
To center (phase) the incoming picture move the joystick in the direction that you want the picture to move.



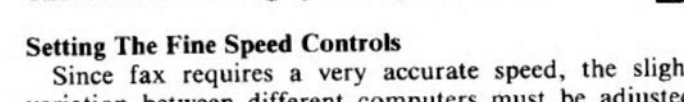
Return joystick to center bottom again and see if picture detail is in from side edges.



Move joystick to top center and wait a second for the scanning to start at the top again.

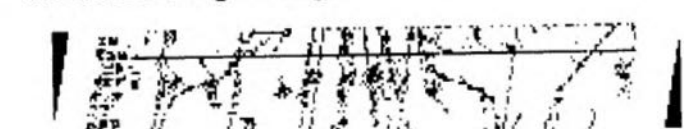


Then resume scanning by moving stick down.



Setting The Fine Speed Controls

Since fax requires a very accurate speed, the slight variation between different computers must be adjusted for. If the picture appears skewed on your computer, adjust the numbers in the BASIC loader. This is an example of the number being too large:



If you were using 120 speed, edit Line 50 and change the last one or two digits (for the example above try decreasing it by eight). Some experimentation will be needed to get it just right.

Type LIST 30-50. I(Adjust these numbers)
 30 F1=645 :REM 60 LPM ADJUST
 Fine speed set 40 F2=898 :REM 90 LPM ADJUST
 50 F3=1024 :REM 120 LPM ADJUST

Write the number down on paper and *RUN* the program. When you have it exactly right, you will probably want to *SAVE* the adjusted copy to avoid having to edit it each time. Note that each speed must be adjusted separately.

Pan Over Picture

Tapping the button when no function is highlighted shows you the picture in memory. Since the picture is much larger than can be displayed, the screen is made into a window which is used to pan over the Hi-Res picture. Move the joystick around to see the rest of the picture. Tapping the button again gets you back to the menu. (Note: if you do this before a picture is received or loaded, you will see a memory start up pattern.)



All the menu functions end with pan over picture. After any function you can tap the button to return to the menu.

Starting And Phasing

WEFAX Receive uses manual start, phasing and stop (abort). Many radios aren't frequency stable enough to make use of the World Meteorological Organization (WMO) remote control signals (300 Hz start, 25s 5% white phasing and 450 Hz stop). To start: select the speed appropriate for the station tuned in. When you hear a low tone followed by a "tweedling" sound, tap the button and move the joystick to center bottom. This is a phasing interval and the picture edge mark.



Phase the white break so that it is on the edge by moving the joystick in that direction.



Reset the scanning to the top.

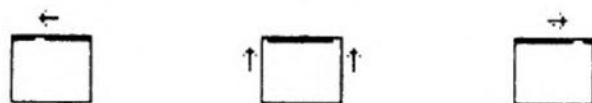


Return joystick to bottom to begin scanning.



Receive Picture

The joystick is used to control the starting and centering of the picture during receive.



The picture you see during receive is only one-ninth the size and resolution. When completed you will see a portion of a much larger picture (see PAN).



Print Picture

The *WEFAX* print routine was designed to work with the Epson MX-80 printer with Graftrax. It is also usable on the Epson FX-80 and RX-80 printers. The Color Computer uses a serial port and thus the printer must have the buffered serial option or an external serial interface. For fastest printout of pictures, the program is set to 9600 Baud; set the printer accordingly. (You could also change the Baud rate in the program, but this would slow down the printout of pictures.) To use another type of printer would require changing the machine code portion of the program.

down button until printing stops. This function does nothing if the printer is off or absent. Turn off the printer when the picture is done to minimize interference during reception.

Change this I to change Baud rate:

55 POKE150, I :REM PRINT=9600 BAUD

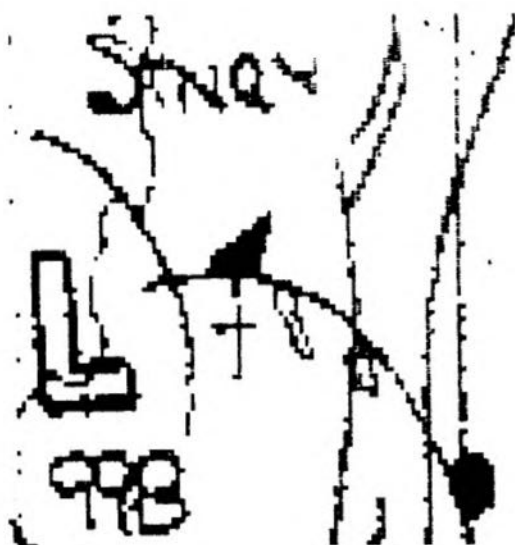
BAUD RATE:	9600	4800	2400	1200	600	300
VALUE:	1	7	18	41	87	180

Reverse Video The Picture

Charts are usually sent on a light background, however, if you get this



select REVERSE VIDEO THE PICTURE to make it look like this.



This function is useful when printing because large black areas squeeze lots of ink out the ribbon and overheat the printer.

(Note, since this function does not change the information content of the picture, you can do it as many times as you want.)

The pictures may be reversed if you are tuned to the wrong sideband for the signal being received. When finished receiving you may want to retune the radio so you won't have to reverse it every time.

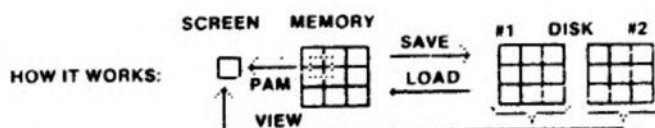
Disk Picture Storage

If WEFAX Receive is run on a computer with a disk system, the following functions will appear in the menu:

```

DISK      ♦VIEW♦ #1 ♦♦♦ #2 ♦♦♦
PICTURE   ♦LOAD♦   ♦♦♦   ♦♦♦
STORAGE   ♦SAVE♦   ♦♦♦   ♦♦♦
    
```

These functions allow you to quickly save the pictures for later viewing. No attempt was made to use tape save since it would be hideously slow. There are three disk functions that can be used. Two pictures may be saved on each disk. First, select one of the functions, then select the disk picture that you want it to access. Note: When you use VIEW, you must select one of the nine screens to view. On many charts the title is in the upper right-hand corner, so select the upper right-hand asterisk. For SAVE and LOAD, you select the whole picture and all the screens in it will be transferred.



program). The pictures will not show up in the disk directory, and if you save too many other things there may not be enough room for the pictures. If there is not enough room the picture will not be saved (you won't see the nine screens flash by). To avoid possible problems, you should probably make up several disks for pictures and copy only the WEFAX program onto each.

Use the following procedure to set up a disk:

- 1) Insert a blank disk in drive 0. Type *DSKIN10,1* (press the ENTER key). This will initialize and erase the disk.
- 2) Replace that disk with one containing the WEFAX. Type *LOAD "WEFAX"* (press ENTER).
- 3) Reinsert the blank (initialized) disk. Type *SAVE "WEFAX"* (press ENTER).

Broadcast Schedule

Coast Guard station NMC at Point Reyes, Calif. broadcasts facsimile pictures on 4344.1, 8680.1, 12728.1 and 17149.3 kHz at the following times:

GMT	Eastern	Pacific	
1500	10 am	7 am	Primary layer depth analysis, experimental period
1715	12:15 pm	9:15 am	Tropical analysis, surface analysis, satellite pictures
2000	3 pm	noon	Fax transmission schedule, 500 millibar contour and maximum wind, satellite pictures
2330	6:30 pm	3:30 pm	Tropical analysis, surface analysis, experimental period
0100	8 pm	5 pm	Surface forecast, sea state forecast, experimental period
0300	10 pm	7 pm	Sea surface temperature analyses, sea and weather forecasts
0500	12 am	9 pm	Surface analysis, extended surface forecast, experimental period



There is enough room on each disk to hold both the two pictures and a few other things (like the WEFAX

ASSEMBLY FILE

by Kevin

Before delving too deeply into the instruction set of the 6809 CPU (that clever little integrated circuit that forms the heart of our computer) we need to have an understanding of the construction of the CPU and to some extent the entire computer.

By now you have probably come across the word 'byte'. One byte is the 'standard' unit of data within our 8 bit computer and has a decimal value of anything from 0 to 255. One 'bit' can be either a binary 1 or binary 0 and if you combine 8 of these 'bits' together you have one byte of data in an 8 bit machine.

8 bits 00000001 = 1 byte

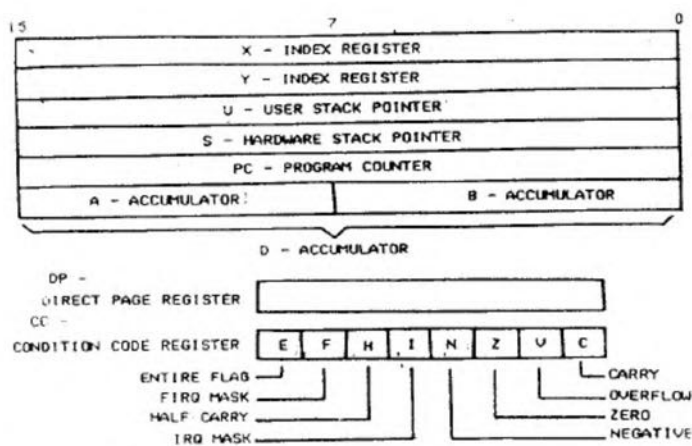
Because the right hand (low order) bit is set (equal to 1 not 0) the decimal value of this byte is 1. Turn to page 184 of the GOING AHEAD WITH EXTENDED COLOR BASIC manual and you will find a complete list of decimal, binary and hex values in the range 0 to 255.

When the 6809 CPU shuffles data between itself and memory it does so one byte at a time along an 8 bit (one byte) data bus. Each bit travels along its own private wire in parallel with its mates. Actual memory locations are referred to along a 16 bit (2 byte, 16 wire) address bus. Thus memory can be 'addressed' in the range 0 to 65535.

16 bits 0000000000000000 = 2 bytes

One memory location is only able to store one byte of data.

Lets now look at a block diagram of the main components of the CPU (fig 1).



You will observe five 16 bit registers (X, Y, U, S, PC), two 8 bit accumulators (A and B), and two 8 bit registers (DP and CC). The A and B accumulators can also be combined to make one large 16 bit accumulator D. These components of the CPU are in effect highly specialised memory locations whose contents are almost continuously changing as they guide and manipulate data around the entire computer.

The five 16 bit registers store the pointer to (address of) individual memory locations. Lets take last months example where we stored a one byte number 35 in memory location 1024. We could have the X register hold the 16 bit number 1024 and the X register thus points to memory address 1024, but address 1024 actually contains the 8 bit number 35. When we dig deeper into indexed addressing the purpose of the X and Y registers will become apparent.

The U and S registers are respectively termed the User and Hardware stack pointers and point to the memory location of the last byte pushed on the respective stack. The U and S stacks are specially designated areas of memory generally used to temporarily store values from any of the CPU registers or accumulators while the CPU carries on with another 'subroutine' which demands the use of those registers/accumulators. Just like stacking plates data can be 'pushed' on top of the stack or 'pulled' off the top of the stack in reverse order.

The PC register is a special purpose register which always points to the memory address of the next instruction to be carried out by the CPU. Thus it's name Program Counter register.

The two accumulators A and B can have thier data merged to make one 16 bit accumulator D. Generally the accumators are used to manipulate data between memory locations or to carry out Arithmetic or Logical operations on data. Remember last month we first loaded accumulator A with the one byte value 35 and then stored it to memory location 1024.

We will conveniently ignore the Direct Page DP register for now, suffice to say we will normally leave it cleared (set to 0).

Finally we come to the CC (Condition Code) register. Each of the 8 bits of the CC register is used as a special flag to the effects of a CPU operation and can be tested for and used to trigger branches to subroutines within your program.

Try to understand the programming architecture of the 6809 and next month we will take a look at it's instruction set.



Staging The Final Scene Of The 51-Column Screen

By R. Bartly Betts

With Programs By Chris Bone

The last of the 51-column text generator is included in this column. From now on you will not be limited to a 32-column Color Computer screen. In fact, to celebrate the conclusion of a new screen text routine, this month's column is dedicated to printing, on and off the screen.

First, just to make changing to the new format easier, the demonstrations in this month's column are for a machine language program that finds BASIC *PRINT@* and *TAB* command parameters to aid you in converting them to the new screen format.

As usual, the listing for the complete 51-column display is at the end of the article. The complete listing is included for any of you who have just joined us, so you do not need to do any appending to the previous listings to get the complete program.

We have received a number of good solutions to the previous challenge to fill a display screen with alpha characters and convert all 'A' characters to asterisks. I will publish some of the best solutions and the authors' names next month. I am sure it seems like a long time between the challenge and the recognition, but the columns have to be written a couple of months before they are published and quite a time delay is involved.

Those who submitted programs have heard from me by now, the rest of you will have to wait another month.

The Groundwork

Now, about a BASIC *TAB* and *PRINT@* finder. In order to create a program to make changes in a BASIC program, there are a few things we must first know:

- 1) Where does the BASIC program begin in memory?
- 2) Where does the BASIC program end in memory?
- 3) What are the codes that pinpoint the data to be translated?

Luckily, these questions can be easily answered. First, memory locations 25 and 26 contain the pointer to the beginning of a BASIC program. To find where any BASIC program begins, multiply the value in Location 25 by 256 and add the value of Location 26.

Memory locations 27 and 28 contain the pointers for the end of a BASIC program, in the same manner. If you have not already tried something similar, enter the following BASIC program to have a look through any program in memory.

First, load a BASIC program, then using line numbers that do not conflict with your program, type and run the following listing:

```
0 A=PEEK(25)
10 B=PEEK(26)
20 L=A*256+B
30 A=PEEK(27)
40 B=PEEK(28)
50 E=A*256+B
60 FOR X=L TO L+32
70 LN$=LN$+CHR$(PEEK(X))
80 NEXT X
90 PRINT LN$;
100 LN$=""
110 L=X
120 IF L>E THEN END ELSE 60
```

The program is rather simple, but it prints out a BASIC program, from beginning to end, in the form in which it resides in memory. The tokens for the BASIC key words are not translated and do appear as graphics characters. Running it gives you an idea of what the *FIND* program deals with.

The Requirements

Your translator program also needs a routine that looks through the BASIC program. As well, it needs the capability to know when it finds a *PRINT* or a *TAB* command and to print out the parameters it finds. The internal BASIC code for *PRINT* is 135 decimal or 87 Hex, and the ASC code for the '@' symbol is 64 decimal or 40 Hex. The internal BASIC code for *TAB* is 164 decimal or A4 Hex.

You need to know that the lines of the new 51-column screen contain 1.6 times as many characters as the old 32-column screen (51 divided by 32).

The following listing is the source code for a program that will search through a BASIC program's memory and locate *PRINT@* or *TAB* commands. Once you have entered and assembled it, you can use it on any BASIC program. It resides in a memory graphics page and so does not interfere with any BASIC program, regardless of length.

The Rest Is Up To You

You will notice the program is just a bare minimum. We have done this because the purpose of the column is to teach assembly language programming. The program provides a base upon which you can build. Some of the things you may wish to add are:

- 1) A conversion to do the new *TAB* or *PRINT@* calculations.
- 2) An option to send the printout to either the screen or a printer.
- 3) A routine that recognizes when the new number has more digits than the old number, open up the BASIC program for insertion and put in the new number.

Try adding to the program according to your skills. Again, Chris and I would be glad to see your results. To aid your endeavors, you might find it helpful to know that replacing a value of 254 in memory location 111 (decimal) or 6F (Hex) causes characters to be sent to the printer rather than the screen. The normal value of Location 111 is zero.

To run the program as it is, type in the following listing and save it under the name *FIND/SOR*, then assemble it under the name *FIND/BIN*. Load in the BASIC program you wish to change, type *LOADM "FIND/BIN* and *ENTER*. Then type *EXEC &HE00* *ENTER*. All of the BASIC line numbers containing the two target commands

are be printed to the screen along with the values of the parameters.

The Print Routine

This program introduces an assembly concept I am sure you will find very useful. Lines 00260, 00290, 00330, 00400 and 00450 make use of built-in ROM routines to print characters to the video display. Calling subroutine \$A002 prints whatever character is currently in register A to the video screen. Calling subroutine \$BDCC prints the two characters contained in register D to the screen. Remember that register D is a 16-bit register composed to registers A and B.

The use of these ROM routines is very simple. Load the appropriate register and jump to the subroutine. When the character or characters are printed, your program continues execution where it left off. The 51-column program in this article makes use of a number of these subroutines and many will be introduced in later articles.

The second part of the print function that needs explaining is found in the last five lines of Listing 2. FCC and FCB are used to set up a table of characters to be printed.

FCC (Form Constant Character) allows you to use any character as a delimiter to establish the beginning and

end of the table. In this case, the slash (/) is used as the delimiter. All characters between the slashes, including the spaces, are printed. The delimiter characters must, of course, not be included in the text.

FCB (Form Constant Byte) is used to establish a character that ends the print routine (lines 00430 to 00460). Notice that Line 440 does a comparison to see if the print character is a zero. If it is, then the routine branches back to the calling routine.

This Is It!

Now, type in and assemble your 51-column program and enjoy a new dimension in the use of your Color Computer. If you feel too finger-weary to type in the complete listing and do not subscribe to RAINBOW ON TAPE, send us \$10 and we will copy and supply the utility on either disk or tape. Be sure to specify which. (In case you haven't noticed, it is cheaper to order this month's RAINBOW ON TAPE. We are not trying to compete with this excellent service.)

All functions of BASIC are preserved in the text generator, including *CLS*.

There is one promise that we have not yet fulfilled. We have had it working in this program, but decided to leave

it out for now. As yet, the text generator does not have a clear to the end of line and clear to the end of the screen function. Try incorporating these functions on your own. Look at how Chris accomplished the *CLS* routine to get ideas. If you have trouble, we will provide the solution in the future.

If you come up with other enhancements, send them along. We won't provide any fortunes, but we can promise fame if you have a good routine.

Next month, we will discuss the use of ROM routines some more and continue with information on assembler commands.

If you need to contact Chris or me, please write to:

R. Bartly Betts/Chris Bone
2251 Lipscomb
Fort Worth, TX 76110
Phone (817) 924-3725
(no collect calls, please!)

51-Column Text Generator Assemble as "TEX51". To execute, type:

```
CLEAR1,&H7CC1: CLEAR 300
LOADM (OR CLOADM) "TEXT51"
ENTER
EXEC &H7CC2
```

Listing 1: (SCREEN 51)

```
7CC2          00010      ORG      $7CC2  *USE THIS LINE FOR 32K
              00020 *    ORG      $3CC2  *USE THIS LINE FOR 16K
              00030 START LDX      #GO    *CHARACTER PRINT ROUTINE
7CC2 8E      7D58      00040      LDD      $168  *OLD RAM HOOK
7CC5 FC      0168      00050      STX      $168  *SAVE NEW HOOK
7CC8 BF      0168      00060      STD      RETURN+1 *SET UP RETURN FROM PRINT
7CCB FD      7DB7      00070      LDX      #PARSE  *PARSE ROUTINE INTERCEPT
7CCE 8E      7CE1      00080      LDD      $A9   *OLD PARSE ROUTINE
7CD1 DC      A9        00090      STX      $A9   *SAVE NEW HOOK
7CD3 9F      A9        00100      STD      PRET+1 *SET UP RETURN
7CD5 FD      7CF8      00110      LDX      $BA   *TOP OF VIDEO PAGE
7CD8 9E      BA        00120      STX      POS   *CURSOR POSITION
7CDA BF      7FFC      00130      CLR      BIT   *LEFT COLUMN
7CDD 7F      7FFA      00140      RTS
7CE0 39      00150 *****
              00160 *A=CURRENT BYTE FROM
              00170 *BASIC PROGRAM
7CE1 81      9E        00180 PARSE  CMPA   #$9E   *TOKEN FOR CLS?
7CE3 26      15        00190      BNE   NOCLS *NOT CLS
7CE5 34      16        00200      PSHS  D,X   *SAVE X AND D
7CE7 9E      BA        00210      LDX   $BA   *TOP OF SCREEN
7CE9 BF      7FFC      00220      STX   POS   *SAVE AS CURSOR POS
              00230      LDD   #$FFFF *WHITE
7CEC CC      FFFF      00240 PCLS1  STD   ,X++  *WHITE OUT
7CEF ED      81        00250      CHPX  $B7   *SCREEN TO BOTTOM
7CF1 9C      B7        00260      BNE   PCLS1
7CF3 26      FA        00270      PULS  D,X   *RESTORE X AND D
7CF5 35      16        00280      JMP   $1000 *RETURN TO PARSE
7CF7 7E      1000      00290 NOCLS  CMPA   #$87   *PRINT?
7CFA 81      87        00300      BNE   PRINT *NOT A PRINT
7CFC 26      05        00310      COM   PATFL *SET PRINT@ FLAG
7CFE 73      7FFF      00320      BRA   PRET  *RETURN
7D01 20      F4        00320
```

7D03	81	40	00330	PRINT	CMPA	#64	*IS IT AN @ SYMBOL
7D05	27	09	00340		BEQ	PRINAT	*IF IT IS GOTO PRINT AT ROUTINE
7D07	81	20	00350		CMPA	#32	*SPACE (SKIP SPACE BETWEEN PRINT AND AT)
7D09	27	EC	00360		BEQ	PRET	*RETURN IF SPACE
7D0B	7F	7FFF	00370		CLR	PATFL	*NOT SPACE SO RESET PRINT FLAG
7D0E	20	E7	00380		BRA	PRET	*RETURN
7D10	7D	7FFF	00390	PRINAT	TST	PATFL	*WE GOT AN AT SO ARE WE PRINTING?
7D13	27	E2	00400		BEQ	PRET	*IF NOT RETURN
7D15	7F	7FFF	00410		CLR	PATFL	*RESET FLAG
7D18	34	16	00420		PSHS	D,X	*SAVE D AND X
7D1A	86	20	00430		LDA	#\$20	*SPACE
7D1C	17	009A	00440		LBSR	LETTER	*BLANK OUT CURSOR
7D1F	BD	B3E4	00450		JSR	\$B3E4	*CALCULATE PRINT AT POSITION
7D22	1083	04C8	00460		CHPD	#\$4C8	*PAST END OF SCREEN
7D26	1024	3720	00470		LBHS	\$B44A	*FC ERROR IF SO
7D2A	1083	0032	00480	CCAL1	CMPD	#50	*CALCULATE
7D2E	23	08	00490		BLS	CCAL2	*CURSOR POSITION
7D30	83	0033	00500		SUBD	#51	*DOWN 256 BYTES FOR EACH 51 IN THE
7D33	7C	7FFF	00510		INC	PATFL	*PRINT AT LOCATION
7D36	20	F2	00520		BRA	CCAL1	
7D38	86	05	00530	CCAL2	LDA	#5	*AND 5 BITS FOR EACH ONE LEFT OVER
7D3A	3D		00540		MUL		*CALCULATE # OF BITS
7D3B	34	04	00550		PSHS	B	*SAVE # OF BITS
7D3D	C4	07	00560		ANDB	#7	*SAVE LSN
7D3F	F7	7FFA	00570		STB	BIT	*SAVE AS BIT COUNT
7D42	B6	7FFF	00580		LDA	PATFL	*GET # LINES DOWN
7D45	35	04	00590		PULS	B	*GET # BYTES ACROSS * 8
7D47	54		00600		LSRB		*DIVIDE
7D48	54		00610		LSRB		* BY
7D49	54		00620		LSRB		* 8
7D4A	D3	BA	00630		ADDD	\$BA	*ADD SCREEN OFFSET
7D4C	FD	7FFC	00640		STD	POS	*SAVE CURSOR POSITION
7D4F	7F	7FFF	00650		CLR	PATFL	*RESET FLAG
7D52	35	16	00660		PULS	D,X	*GET BACK D AND X
7D54	86	20	00670		LDA	#32	*SPACE TO FOOL BASIC INTO THINKING
7D56	20	9F	00680		BRA	PRET	*THERE WAS TO @ SYMBOL
7D58	0D	6F	00690	GO	TST	\$6F	*FROM LAST MONTH
7D5A	1026	0058	00700		LBNE	RETURN	
7D5E	34	16	00710		PSHS	A,B,X	
7D60	81	08	00720		CMPA	#8	
7D62	26	13	00730		BNE	NOTBS	
7D64	17	00FD	00740		LBSR	BACK	
7D67	86	20	00750		LDA	#32	
7D69	8D	4E	00760		BSR	LETTER	
7D6B	86	20	00770		LDA	#32	*EXTRA SPACE TO ERASE CURSOR
7D6D	8D	4A	00780		BSR	LETTER	
7D6F	17	00F2	00790		LBSR	BACK	*EXTRA BACKUP TO FIX EXTRA SPACE
7D72	17	00EF	00800		LBSR	BACK	
7D75	20	36	00810		BRA	RET	
7D77	81	0C	00820	NOTBS	CMPA	#12	
7D79	26	13	00830		BNE	NOTCL	
7D7B	9E	BA	00840		LDX	\$BA	
7D7D	BF	7FFC	00850		STX	POS	
7D80	7F	7FFA	00860		CLR	BIT	
7D83	CC	FFFF	00870	CLS	LDD	#\$FFFF	
7D86	ED	81	00880	CLLOO	STD	,X++	
7D88	9C	B7	00890		CMPX	\$B7	
7D8A	26	FA	00900		BNE	CLLOO	
7D8C	20	1F	00910		BRA	RET	
7D8E	8D	29	00920	NOTCL	BSR	LETTER	
7D90	BE	7FFC	00930		LDX	POS	
7D93	9C	B7	00940		CMPX	\$B7	
7D95	25	16	00950		BLO	RET	
7D97	30	89 FF00	00960		LEAX	-256,X	
7D9B	BF	7FFC	00970		STX	POS	
7D9E	9E	BA	00980		LDX	\$BA	
7DA0	EC	89 0100	00990	SCLOOP	LDD	256,X	
7DA4	ED	81	01000		STD	,X++	
7DA6	BC	7FFC	01010		CMPX	POS	
7DA9	26	F5	01020		BNE	SCLOOP	
7DAB	20	D6	01030		BRA	CLS	

One-Liner

This is a skiing game using the right joystick. Move your skis left or right as you go down the mountain. Try to pass between the flags. The game ends when you knock down 10 flags.

The listing:

```

1 S=RND(15):PRINT@S+487,"^ ^";:F
ORX=1TO9:Z=INT(JOYSTK(0)/22)-1:R
=R+Z:PRINT@507,G"yyyy"P;:PRINT@2
00+R,"#";:PLAY"T15A":NEXT:IFR=S
THENPLAY"B":P=P+5:GOTO1ELSEP=P-1
:IFR+1=S ORR-1=S THENPLAY"FD":PR
INT@199+R,"_";:G=G+1:IFG=10THENP
RINTG ELSE1ELSE1

```

Kevin Derby
Lubbock, TX

7DAD 86	5F	01040	RET	LDA	#95	*CURSOR SYMBOL (UNDERLINE)					
7DAF 8D	08	01050		BSR	LETTER	*PRINT CURSOR					
7DB1 17	00B0	01060		LBSR	BACK	*BACK UP TO CORRECT PRINT POS					
7DB4 35	96	01070		PULS	A,B,X,PC						
7DB6 7E	1000	01080	RETURN	JMP	\$1000		7E46 30	01	01750	LEAX	1,X
7DB9 81	0D	01090	LETTER	CMPA	#\$0D		7E48 C0	08	01760	SUBB	#8
7DBB 26	11	01100		BNE	LETT2		7E4A C1	07	01770	BITSV	CMPB #7
7DBD 86	20	01110		LDA	#\$20		7E4C 26	0F	01780	BNE	BITDNI
7DBF 8D	F8	01120		BSR	LETTER		7E4E 1E	10	01790	EXG	X,D
7DC1 17	00A0	01130		LBSR	BACK		7E50 C1	1F	01800	CMPB	#\$1F
7DC4 7F	7FFD	01140		CLR	POS+1		7E52 26	07	01810	BNE	BITDN
7DC7 7C	7FFC	01150		INC	POS		7E54 4C		01820	INCA	
7DCA 7F	7FFA	01160		CLR	BIT		7E55 5F		01830	CLRB	
7DCD 39		01170		RTS			7E56 1E	01	01840	EXG	D,X
7DCE C6	04	01180	LETT2	LDB	#4		7E58 5F		01850	CLRB	
7DD0 F7	7FFE	01190		STB	BYTE		7E59 20	02	01860	BRA	BITDNI
7DD3 80	20	01200		SUBA	#\$20		7E5B 1E	01	01870	BITDN	EXG D,X
7DD5 3D		01210		MUL			7E5D F7	7FFA	01880	BITDNI	STB BIT
7DD6 C3	7E8C	01220		ADDD	#TABLE		7E60 BF	7FFC	01890	STX	POS
7DD9 1F	02	01230		TFR	D,Y		7E63 39		01900	RTS	
7DDB BE	7FFC	01240		LDX	POS		7E64 34	04	01910	BACK	PSHS B
7DDE A6	A0	01250	LOOP	LDA	,Y+		7E66 F6	7FFA	01920	LDB	BIT
7DE0 34	02	01260		PSHS	A		7E69 BE	7FFC	01930	LDX	POS
7DE2 8A	0F	01270		ORA	#\$0F		7E6C C0	05	01940	SUBB	#5
7DE4 8D	16	01280		BSR	SLICE		7E6E 2A	14	01950	BPL	BACKSP
7DE6 35	02	01290		PULS	A		7E70 CB	08	01960	ADDB	#8
7DE8 48		01300		ASLA			7E72 30	1F	01970	LEAX	-1,X
7DE9 48		01310		ASLA			7E74 1E	10	01980	EXG	X,D
7DEA 48		01320		ASLA			7E76 C1	FF	01990	CMPB	#\$FF
7DEB 48		01330		ASLA			7E78 26	08	02000	BNE	BACKS1
7DEC 8A	0F	01340		ORA	#\$0F		7E7A C6	1F	02010	LDB	#\$1F
7DEE 8D	0C	01350		BSR	SLICE		7E7C 1E	10	02020	EXG	X,D
7DF0 7A	7FFE	01360		DEC	BYTE		7E7E C6	02	02030	LDB	#2
7DF3 26	E9	01370		BNE	LOOP		7E80 20	02	02040	BRA	BACKSP
7DF5 30	89 FFO0	01380		LEAX	-256,X		7E82 1E	10	02050	BACKS1	EXG X,D
7DF9 8D	42	01390		BSR	FORW		7E84 BF	7FFC	02060	BACKSP	STX POS
7DFB 39		01400		RTS			7E87 F7	7FFA	02070	STB	BIT
7DFC F6	7FFA	01410	SLICE	LDB	BIT		7E8A 35	84	02080	PULS	B,PC
7DFF 5C		01420		INCB			7E8C	FFFF	02090	TABLE	FDB \$FFFF
7E00 F7	7FFB	01430		STB	BIT+1		7E8E	FFFF	02100	FDB	\$FFFF
7E03 34	02	01440		PSHS	A		7E90	DDDD	02110	FDB	\$DDDD
7E05 CC	F800	01450		LDD	#\$F800		7E92	DFDF	02120	FDB	\$DFDF
7E08 8A	08	01460		ORA	#\$8		7E94	55FF	02130	FDB	\$55FF
7E0A FD	7FF8	01470		STD	MASK		7E96	FFFF	02140	FDB	\$FFFF
7E0D 35	02	01480		PULS	A		7E98	9909	02150	FDB	\$9909
7E0F C6	FF	01490		LDB	#\$FF		7E9A	099F	02160	FDB	\$099F
7E11 7A	7FFB	01500	BITTST	DEC	BIT+1		7E9C	B17B	02170	FDB	\$B17B
7E14 27	0E	01510		BEQ	SAVE		7E9E	D1BF	02180	FDB	\$D1BF
7E16 1A	01	01520		ORCC	#\$1		7EA0	F32D	02190	FDB	\$F32D
7E18 46		01530		RORA			7EA2	B4CF	02200	FDB	\$B4CF
7E19 56		01540		RORB			7EA4	B55B	02210	FDB	\$B55B
7E1A 1C	FE	01550		ANDCC	#\$FE		7EA6	25AF	02220	FDB	\$25AF
7E1C 76	7FF8	01560		ROR	MASK		7EA8	DBFF	02230	FDB	\$DBFF
7E1F 76	7FF9	01570		ROR	MASK+1		7EAA	FFFF	02240	FDB	\$FFFF
7E22 20	ED	01580		BRA	BITTST		7EAC	DB77	02250	FDB	\$DB77
7E24 34	06	01590	SAVE	PSHS	D		7EAE	7BDF	02260	FDB	\$7BDF
7E26 FC	7FF8	01600		LDD	MASK		7EB0	BDEE	02270	FDB	\$BDEE
7E29 AA	84	01610		ORA	,X		7EB2	EDBF	02280	FDB	\$EDBF
7E2B EA	01	01620		ORB	1,X		7EB4	F690	02290	FDB	\$F690
7E2D A7	84	01630		STA	,X		7EB6	96FF	02300	FDB	\$96FF
7E2F E7	01	01640		STB	1,X		7EB8	FBB1	02310	FDB	\$FBB1
7E31 35	06	01650		PULS	D		7EBA	BBFF	02320	FDB	\$BBFF
7E33 A4	84	01660		ANDA	,X		7EBC	FFFF	02330	FDB	\$FFFF
7E35 E4	01	01670		ANDB	1,X		7EBE	9DBF	02340	FDB	\$9DBF
7E37 ED	84	01680		STD	,X		7EC0	FFFO	02350	FDB	\$FFFO
7E39 30	88 20	01690		LEAX	32,X		7EC2	FFFF	02360	FDB	\$FFFF
7E3C 39		01700		RTS			7EC4	FFFF	02370	FDB	\$FFFF
7E3D F6	7FFA	01710	FORW	LDB	BIT		7EC6	FDDF	02380	FDB	\$FDDF
7E40 CB	05	01720		ADDB	#5		7EC8	FDDB	02390	FDB	\$FDDB
7E42 C1	07	01730		CMPB	#7		7ECA	B77F	02400	FDB	\$B77F
7E44 23	04	01740		BLS	BITSV		7ECC	9640	02410	FDB	\$9640

7ECE	269F	02420	FDB	\$269F	7F5C	ODDD	03130	FDB	\$ODDD
7ED0	D9DD	02430	FDB	\$D9DD	7F5E	DDDF	03140	FDB	\$DDDF
7ED2	DD8F	02440	FDB	\$DD8F	7F60	6666	03150	FDB	\$6666
7ED4	96ED	02450	FDB	\$96ED	7F62	669F	03160	FDB	\$669F
7ED6	B70F	02460	FDB	\$B70F	7F64	6666	03170	FDB	\$6666
7ED8	96E9	02470	FDB	\$96E9	7F66	699F	03180	FDB	\$699F
7EDA	E69F	02480	FDB	\$E69F	7F68	6666	03190	FDB	\$6666
7EDC	D950	02490	FDB	\$D950	7F6A	006F	03200	FDB	\$006F
7EDE	DDDF	02500	FDB	\$DDDF	7F6C	6699	03210	FDB	\$6699
7EE0	071E	02510	FDB	\$071E	7F6E	966F	03220	FDB	\$966F
7EE2	E69F	02520	FDB	\$E69F	7F70	6660	03230	FDB	\$6660
7EE4	DB71	02530	FDB	\$DB71	7F72	DDDF	03240	FDB	\$DDDF
7EE6	669F	02540	FDB	\$669F	7F74	OEC9	03250	FDB	\$OEC9
7EE8	OEE0	02550	FDB	\$OEE0	7F76	370F	03260	FDB	\$370F
7EEA	B77F	02560	FDB	\$B77F	7F78	8BBB	03270	FDB	\$8BBB
7EEC	9669	02570	FDB	\$9669	7F7A	B88F	03280	FDB	\$888F
7EEE	669F	02580	FDB	\$669F	7F7C	F77B	03290	FDB	\$F77B
7EFO	9668	02590	FDB	\$9668	7F7E	BDDF	03300	FDB	\$BDDF
7EF2	EDBF	02600	FDB	\$EDBF	7F80	1DDD	03310	FDB	\$1DDD
7EF4	FDDF	02610	FDB	\$FDDF	7F82	DD1F	03320	FDB	\$DD1F
7EF6	DDFF	02620	FDB	\$DDFF	7F84	B55F	03330	FDB	\$B55F
7EF8	FDDF	02630	FDB	\$FDDF	7F86	FFFF	03340	FDB	\$FFFF
7EFA	DDBF	02640	FDB	\$DDBF	7F88	FFFF	03350	FDB	\$FFFF
7EFC	EDB7	02650	FDB	\$EDB7	7F8A	FF0F	03360	FDB	\$FF0F
7EFE	BDEF	02660	FDB	\$BDEF	7F8C	BDFE	03370	FDB	\$BDFE
7F00	FF0F	02670	FDB	\$FF0F	7F8E	FFFF	03380	FDB	\$FFFF
7F02	OFFF	02680	FDB	\$OFFF	7F90	FF1E	03390	FDB	\$FF1E
7F04	7BDE	02690	FDB	\$7BDE	7F92	868F	03400	FDB	\$868F
7F06	DB7F	02700	FDB	\$DB7F	7F94	7771	03410	FDB	\$7771
7F08	96ED	02710	FDB	\$96ED	7F96	661F	03420	FDB	\$661F
7FOA	BFBF	02720	FDB	\$BFBF	7F98	FF87	03430	FDB	\$FF87
7FOC	9642	02730	FDB	\$9642	7F9A	778F	03440	FDB	\$778F
7FOE	478F	02740	FDB	\$478F	7F9C	EEE8	03450	FDB	\$EEE8
7F10	9660	02750	FDB	\$9660	7F9E	668F	03460	FDB	\$668F
7F12	666F	02760	FDB	\$666F	7FA0	FF96	03470	FDB	\$FF96
7F14	3551	02770	FDB	\$3551	7FA2	079F	03480	FDB	\$079F
7F16	661F	02780	FDB	\$661F	7FA4	DAB1	03490	FDB	\$DAB1
7F18	9677	02790	FDB	\$9677	7FA6	BBBF	03500	FDB	\$BBBF
7F1A	769F	02800	FDB	\$769F	7FA8	F966	03510	FDB	\$F966
7F1C	1AAA	02810	FDB	\$1AAA	7FAA	8E8F	03520	FDB	\$8E8F
7F1E	AA1F	02820	FDB	\$AA1F	7FAC	7716	03530	FDB	\$7716
7F20	0771	02830	FDB	\$0771	7FAE	666F	03540	FDB	\$666F
7F22	770F	02840	FDB	\$770F	7FB0	DF9D	03550	FDB	\$DF9D
7F24	0771	02850	FDB	\$0771	7FB2	DD8F	03560	FDB	\$DD8F
7F26	777F	02860	FDB	\$777F	7FB4	EFEE	03570	FDB	\$EFEE
7F28	9674	02870	FDB	\$9674	7FB6	E69F	03580	FDB	\$E69F
7F2A	669F	02880	FDB	\$669F	7FB8	F764	03590	FDB	\$F764
7F2C	6660	02890	FDB	\$6660	7FBA	166F	03600	FDB	\$166F
7F2E	666F	02900	FDB	\$666F	7FBC	3BBB	03610	FDB	\$3BBB
7F30	8DDD	02910	FDB	\$8DDD	7FBE	BB1F	03620	FDB	\$BB1F
7F32	DD8F	02920	FDB	\$DD8F	7FC0	FF60	03630	FDB	\$FF60
7F34	CEEE	02930	FDB	\$CEEE	7FC2	666F	03640	FDB	\$666F
7F36	E69F	02940	FDB	\$E69F	7FC4	FF16	03650	FDB	\$FF16
7F38	6533	02950	FDB	\$6533	7FC6	666F	03660	FDB	\$666F
7F3A	356F	02960	FDB	\$356F	7FC8	FF96	03670	FDB	\$FF96
7F3C	7777	02970	FDB	\$7777	7FCA	669F	03680	FDB	\$669F
7F3E	770F	02980	FDB	\$770F	7FCC	F166	03690	FDB	\$F166
7F40	6006	02990	FDB	\$6006	7FCE	177F	03700	FDB	\$177F
7F42	666F	03000	FDB	\$666F	7FD0	F866	03710	FDB	\$F866
7F44	6224	03010	FDB	\$6224	7FD2	8EEF	03720	FDB	\$8EEF
7F46	446F	03020	FDB	\$446F	7FD4	FF16	03730	FDB	\$FF16
7F48	9666	03030	FDB	\$9666	7FD6	777F	03740	FDB	\$777F
7F4A	669F	03040	FDB	\$669F	7FD8	FF07	03750	FDB	\$FF07
7F4C	1661	03050	FDB	\$1661	7FDA	OE0F	03760	FDB	\$OE0F
7F4E	777F	03060	FDB	\$777F	7FDC	BB1B	03770	FDB	\$BB1B
7F50	9666	03070	FDB	\$9666	7FDE	BBBF	03780	FDB	\$BBBF
7F52	25AF	03080	FDB	\$25AF	7FE0	FF66	03790	FDB	\$FF66
7F54	1661	03090	FDB	\$1661	7FE2	669F	03800	FDB	\$669F
7F56	356F	03100	FDB	\$356F	7FE4	FF66	03810	FDB	\$FF66
7F58	9679	03110	FDB	\$9679	7FE6	699F	03820	FDB	\$699F
7F5A	E69F	03120	FDB	\$E69F	7FE8	FF66	03830	FDB	\$FF66

7FEA	606F	03840	FDB	\$606F	7FF8	03910 MASK	RMB	2
7FEC	FF69	03850	FDB	\$FF69	7FFA	03920 BIT	RMB	2
7FEE	966F	03860	FDB	\$966F	7FFC	03930 POS	RMB	2
7FF0	F666	03870	FDB	\$F666	7FFE	03940 BYTE	RMB	1
7FF2	8E9F	03880	FDB	\$8E9F	7FFF	03950 PATFL	RMB	1
7FF4	F0E	03890	FDB	\$F0E		0000 03960	RMB	1
7FF6	DB0F	03900	FDB	\$DB0F			END	
					00000	TOTAL ERRORS		

Listing 2: (FIND)

```

0E00          00010      ORG      $E00
0E00 9E 19      00020 START  LDX      25      *GET BEGINNING OF BASIC PROGRAM
0E02 1F 12      00030 LOOP1  TFR      X,Y      *TRANSFERE TO REGISTER Y
0E04 A6 80      00040 LOOP   LDA      ,X+      *GET FIRST CHARACTER IN PROGRAM
0E06 81 87      00050      CMPA     #135     *IS IT A PRINT CODE?
0E08 27 0D      00060      BEQ      GOTONE  *YES, GO CHECK FOR @
0E0A 81 A4      00070      CMPA     #164     *IS IT TAB CODE?
0E0C 27 31      00080      BEQ      TAB      *YES, GO TO PRINT ROUTINE
0E0E AC A4      00090 LOOP2  CMPX     ,Y      *ARE WE AT THE END OF CURRENT LINE?
0E10 25 F2      00100      BLO     LOOP      *NO, CHECK NEXT CHARACTER
0E12 AE A4      00110      LDX      ,Y      *YES, GET NEXT LINE
0E14 26 EC      00120      BNE     LOOP1    *START LOOKING IN NEXT LINE
0E16 39          00130      RTS
0E17 A6 80      00140 GOTONE  LDA      ,X+      *GET FIRST CHARACTER AFTER PRINT CODE
0E19 81 40      00150      CMPA     #64      *IS IT @?
0E1B 26 F1      00160      BNE     LOOP2    *NO, GO BACK AND LOOK SOME MORE
0E1D EC 22      00170      LDD     2,Y      *YES, GET LINE NUMBER
0E1F 34 30      00180      PSHS    X,Y      *SAVE X AND Y TO STACK
0E21 BD BDCC    00190      JSR     $BDCC    *PRINT LINE NUMBER TO SCREEN
0E24 8E 0E65    00200      LDX     #PR      *GET LOCATION OF CHARACTERS TO PRINT
0E27 8D 31      00210      BSR     TEXTOU   *GO TO PRINT ROUTINE
0E29 35 30      00220      PULS    X,Y      *RETURN X AND Y FROM STACK
0E2B A6 80      00230 READ   LDA      ,X+      *LOAD A WITH NUMBER TO PRINT
0E2D 81 2C      00240      CMPA     #'      *IS IT A COMMA
0E2F 27 06      00250      BEQ     TRAN     *YES, END OF NUMBER
0E31 AD 9F A002 00260      JSR     [$A002]  *PRINT CHARACTER IN A
0E35 20 F4      00270      BRA     READ     *GO GET NEXT CHARACTER
0E37 86 0D      00280 TRAN   LDA      #13      *LOAD A WITH A SPACE CHARACTER
0E39 AD 9F A002 00290      JSR     [$A002]  *GO PRINT A SPACE
0E3D 20 C5      00300      BRA     LOOP     *GO CHECK REST OF CURRENT LINE
0E3F EC 22      00310 TAB   LDD     2,Y      *GET LINE NUMBER
0E41 34 30      00320      PSHS    X,Y      *SAVE X AND Y TO STACK
0E43 BD BDCC    00330      JSR     $BDCC    *PRINT LINE NUMBER
0E46 8E 0E6D    00340      LDX     #TA      *GET LOCATION OF TAB TEXT
0E49 8D 0F      00350      BSR     TEXTOU   *GO PRINT TAB
0E4B 35 30      00360      PULS    X,Y      *GET X AND Y FROM STACK
0E4D A6 80      00370 READ2  LDA      ,X+      *GET NEXT CHARACTER
0E4F 81 29      00380      CMPA     #' )    *IS IT A RIGHT PAREN?
0E51 27 E4      00390      BEQ     TRAN     *YES, END OF NUMBER
0E53 AD 9F A002 00400      JSR     [$A002]  *NO, GO PRINT CHARACTER
0E57 20 F4      00410      BRA     READ2    *GET NEXT CHARACTER
0E59 39          00420 END   RTS
0E5A A6 80      00430 TEXTOU LDA      ,X+      *LOAD CHARACTER TO PRINT
0E5C 27 06      00440      BEQ     TEXT1    *IF IT IS A 0 THEN QUI
0E5E AD 9F A002 00450      JSR     [$A002]  *PRINT THE CHARACTER
0E62 20 F6      00460      BRA     TEXTOU   *GET NEXT CHARACTER
0E64 39          00470 TEXT1  RTS
0E65          20      00480 PR   FCC      / PRINT /
0E6C          00      00490      FCB     $0
0E6D          20      00500 TA   FCC      / TAB /
0E74          00      00510      FCB     $0
          0000      00520      END
00000 TOTAL ERRORS

```

One-Liner

Guaranteed to drive you up the wall after a minute or so, this program draws a circle that gets larger and smaller over and over!

Michael Rosenberg
Prestonburg, KY

The listing:

April, 1985

```

Ø PCLEAR8:FORX=1TO8:PMODEØ,X:PCL
S:CIRCLE(128,96),X*1Ø:NEXT:FORX=
1TO999999:FORY=1TO8:SOUNDY*2Ø,1:
PMODEØ,Y:SCREEN1,1:NEXTY:FORY=8T
O1STEP-1:SOUNDY*2Ø,1:PMODEØ,Y:SC
REEN1,1:NEXTY,X

```

Part VI

COOKING
WITH
CoCo



By Colin J. Stearman

I love my printer. It prints quickly, it prints letter quality, it draws pictures, I can send it my own character fonts . . . but the darn thing has a parallel port and CoCo has a serial printer output. Sure I can buy a serial interface for it but it's over 25 percent of the cost of the printer alone, and I hate to waste money. The only solution is to design a parallel port for CoCo.

The actual design is easy, but I wanted the software to fully integrate the port into BASIC, allowing me to direct printer output to either the parallel port or the existing serial port; and for good measure I wanted the BASIC to allow easy adjustment of the Baud rate on the serial port.

To achieve all this meant adding initialization code for the parallel port hardware, trapping output destined to go to the serial port and redirecting it to the desired printer port. This month's assembly language listing does all that as well as adding three new BASIC commands. If you do not need this parallel port and are thinking of turning to the next article, two of the new commands apply to the existing serial port also, so maybe you might want to stick around.

But before we get to the software, let's get the hardware built. If you didn't have trouble with the EPROM programmer, this project will be a snap.

Adding The Parallel Port

The object of the construction is to mount a new 6821 PIA (peripheral interface adapter) inside the computer, without making irreversible modifications to the circuit board. I did this by "piggybacking" the new PIA onto U4. The photos of my unit should give you an overall idea of the look of the finished unit.

U4 is an existing PIA used to drive the D/A converter and control the VDG chip. Please note that these modifications refer to the REV E-style motherboard. If you have a later model, your PIA may not be labeled U4 and will have to be identified by the function it performs.

To construct the unit, first gather the following components together:

- 1) 6821 PIA Peripheral Interface Adapter
- 2) Breadboard PCB Radio Shack #276-158
- 3) SN7404 Hex Inverter Radio Shack #276-1802
- 4) 40 Pin IC socket, wire-wrap type
- 5) Thin hook up wire
- 6) Flat ribbon cable, 36 conductor wide
- 7) Centronics-type female plug, ribbon mounting

Items 1, 4, 6 and 7 are not carried by Radio Shack but are available via mail order from Active Electronics, Westboro, Mass. and other sources. The IC socket must be the wire-wrap type.

To assemble the parts, first remove the cover from CoCo and also the RF shield lid inside. Locate U4 (REV E board #), the 6821 on the right as you face CoCo. Gently pry the IC out of its socket, using a small screwdriver or IC puller. Be careful not to damage the pins. Put CoCo to one side as we will now construct the "piggyback" board assembly.

Mount the 40-pin socket to the PCB (printed circuit board, item 3) anywhere convenient, but leave room for the SN7404 near pin 24. Solder all pins on the socket to the PCB, but *do not* cut off the excess.

Take the new 6821 and gently bend pin 24 outward a little so that when the IC is put into the socket, this pin will not enter it. Put the IC in the socket and press it home.

Mount the SN7404 alongside the 6821 near pin 24. Solder all pins to the PCB. Using the hookup wire, connect pins 1, 3, 5, 7, 9 and 11 together and also to pin 20 of the 40-pin socket. Connect pin 14 to pin 1 on the 40-pin socket. Connect pin 13 to pin 24 of the 6821. This is the bent pin not inserted into the socket. Also connect this pin to a length of wire about nine inches long. The other end will be connected later. Connect pin 12 to the 40-pin socket pin 24.

Turn the PCB upside down and cut off the wire-wrap pins from pins 2 through 19 only. Cut them as close as possible to the PCB. The next task is to mount the assembly on top of the 6821 removed from U4.

Locate the 6821 removed earlier from CoCo and carefully bend pin 24 so it points vertically upward. Position the assembly on top of this IC to test for fit. It may be necessary to splay the wire-wrap pins out a little. In order for the finished assembly to fit under the RF shield lid, the remaining wire-wrap pins must be trimmed as short as possible. Gauge how much you can cut from each pin and then trim all to this height.

Now solder the assembly to the 6821, soldering each wire-wrap pin to its respective pin on U4. You should be connecting to U4 pins 1, 20, and 21 through 40. The wire-wrap pin 24 will connect to the upturned pin 24 on U4. This pin will not connect to the socket when U4 is returned to the CoCo motherboard. When soldering the assembly to the back of U4, minimize the amount of solder used so that the IC will still fit

into its socket. Also position the solder joint high on the pins so that the lower part will still fit the socket.

Finally, the ribbon cable must be attached to the assembly. Consult your printer manual and Figure 1. The best approach is to fit the Centronics socket to the ribbon cable to aid in identifying the wire numbers. Most sockets have the numbers molded into them. On the standard parallel interface the wires and their functions are:

Wire	Function
1	Data Strobe (bar)
2	Data Bit 1
3	Data Bit 2
4	Data Bit 3
5	Data Bit 4
6	Data Bit 5
7	Data Bit 6
8	Data Bit 7
9	Data Bit 8
11	Busy
14	Signal Ground

Connect the wires to the pins as indicated in Figure 1. Wire 14 should be connected to any 0V point on the assembly. Finally trim all the excess PCB from the assembly to minimize its size.

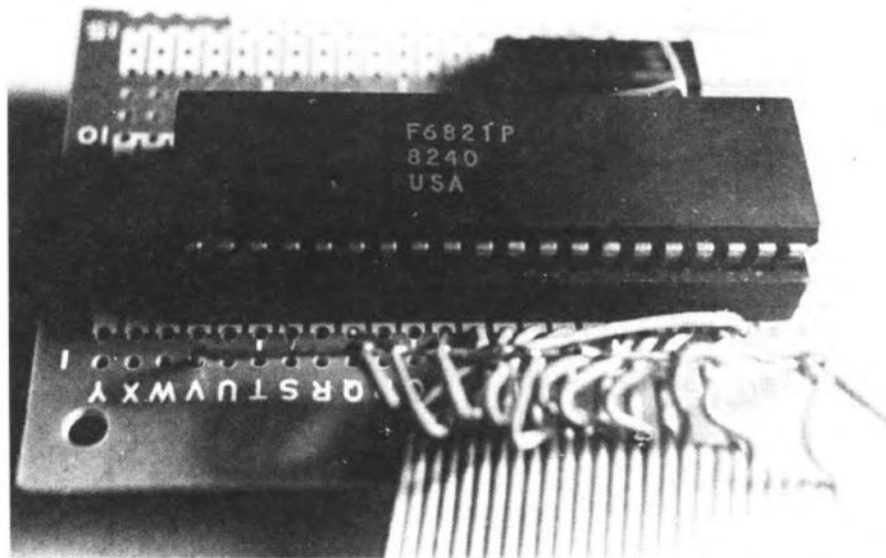
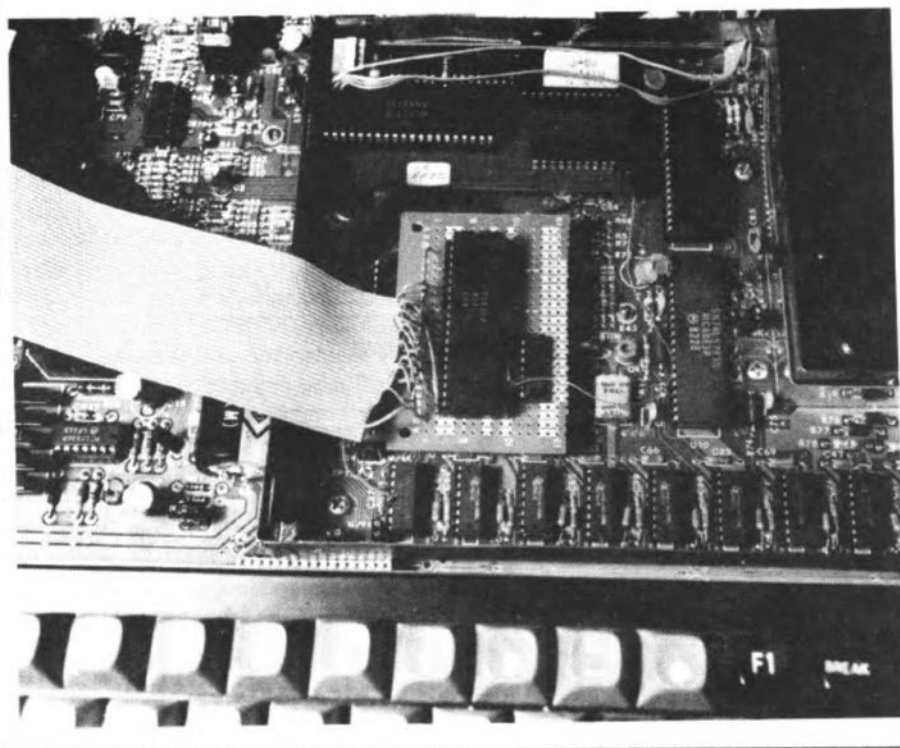
Now mount the finished assembly into CoCo. Press the lower IC gently but firmly into the U4 socket. All pins of the lower 6821 must enter the socket, except for pin 24 which was bent upwards. It's not easy to see that this happens, so inspect the results carefully. The assembly should be firm and quite rigid when installed.

The wire still left unattached must be soldered to the main computer board near the 6809. Cut this wire to a suitable length and attach to the solder point, as indicated in Figure 2. Use a light solder tack to minimize the possibility of damage to the board. This wire picks up address Line 2 to allow the software to distinguish between the two PIAs.

The ribbon cable will head toward the left as you face CoCo. Take the RF shield lid and bend the fingers where the cable is, so the lid can be replaced without pinching the wire. If the assembly is too high to allow the lid to be replaced, either leave it off entirely, or extend the height of the RF shield using some shielding metal from an old TV.

The ribbon cable can be routed out of the computer by doubling it back on itself and running it under the main circuit board. A notch cut in the lower plastic shell underneath the serial and cassette ports will allow the cable to leave the case.

This completes the hardware construc-



tion. We now move on to this month's software additions to the Disk BASIC patch.

The New BASIC Commands

This month we add three new commands, all associated with the printer port. Two apply even if you do not intend building the parallel port, so stick with us.

PARALLEL

Issuing the BASIC command PARALLEL, either directly from the keyboard, or within a program will result in all data destined for the printer being routed out of the new parallel port. In

other words, all *PRINT#-2* statements will output through the parallel port.

The code to drive the parallel port is conditional assembled based upon whether a variable called *PARPNT* is defined or not. Review the paragraph in September's issue for more details on how to include or exclude the code for the parallel port, as desired.

BAUD

This command applies whether or not you have the parallel port. Either way, it establishes the Baud rate of the serial port. If you have the parallel port,

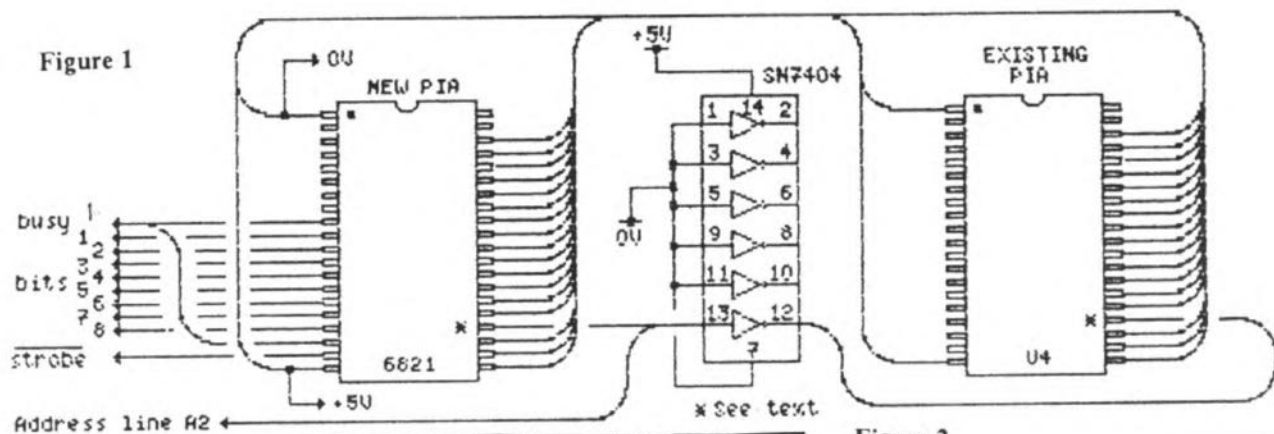
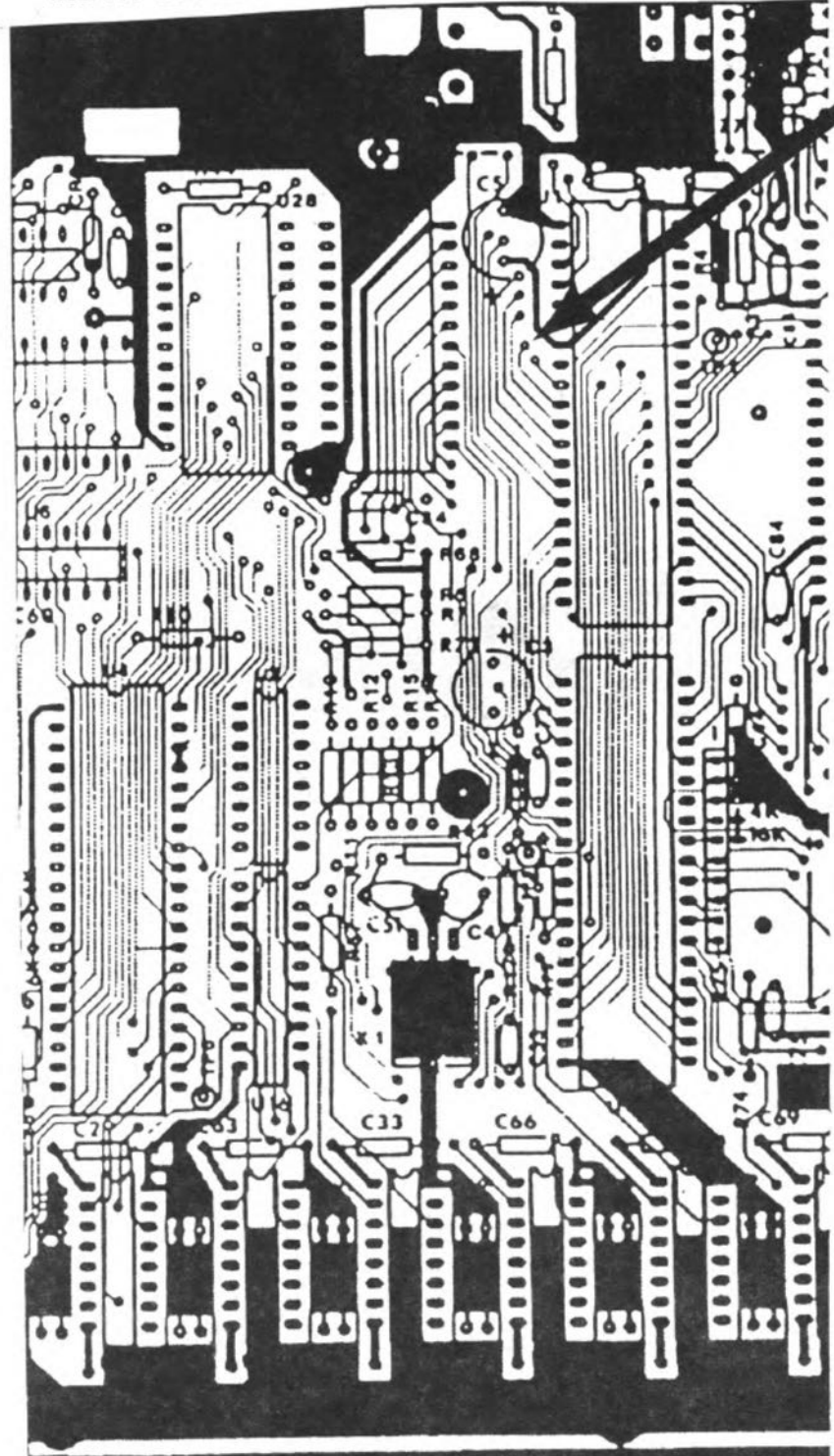


Figure 2



Attach address Line 2 wire from parallel port to this point on Rev 'E' boards.

On other revision boards, locate the trace from Pin 10 on the 6809 micro-processor.

it also activates the serial port so that all *PRINTH-2* commands direct output through the standard serial port. The original serial driver code in the Color BASIC ROM is still used for the serial port.

The syntax for this command is:

BAUD(n)

where n = 300, 600, 1200, 2400, 4800 or 9600.

If you have the parallel port, then CoCo starts up with this activated. If you do not, then the serial port is activated and set at 600 Baud.

LDIR

A simple but useful command which does a normal directory but directs it to the currently active printer port. The directory contains the creation date enhancement, but, of course, does not pause after each 16 lines, as when directed to the screen.

Adding This Month's Code

As last month, use your editor to pull in the source code built up so far. Delete the lines identified with reference numbers 20, 21, 22, and 29. Read and follow the notes at reference Lines 6, 7, 8, 10 and 11 regarding including or excluding the parallel port code.

Go to the end of the listing and delete all the remaining lines from and including *ZZLAST EQU *-1*. Then add the assembly text in Listing 1. When all is set, re-assemble the resulting file and test as you have in previous months.

To test the parallel port, connect it to a printer and try *LLISTing* a BASIC program or run some other program which

has printer output. If it does not work, but the computer works otherwise, double check your wiring on the new PIA, especially around the ribbon cable connection point. It's very easy to miscount the wires.

A Final Point

All BASIC programs will have no trouble sending output to the parallel port. However, you may have trouble with some machine language programs. If they use the serial port in the Color BASIC ROM and do not "mess" with the hooks in RAM, the port should work alright. If the program has Baud rate control, set it to 110 or 120 and this will activate the parallel port; 300 or higher will activate the serial port.

If you have FHL FLEX then you can use the parallel port driver routine described in the FLEX manual. The reason that the BUSY line goes to both pins 9 and 19 on the new PIA is specifically to accommodate the approach these routines use to detect the printer busy con-

The listing:

```

1007 OPT LIS
1008 *****
1009 * PATCH #4 to RSDOS (C)1984 Colin Stearman *
1010 *****
1011 *
1012 * "BAUD" COMMAND CODE
1013 * SYNTAX IS BAUD(N) WHERE N =
1014 * 300,600,1200,2400,4800,9600
1015 *
DC5E DE 1016 BDCNST FCB #BE,057,029,012,06,01 300,600,1200,2400
1017 * 4800,9600 BAUD CONSTANTS
1018 *
1019 *
DC44 BDB262 1100 BAUD JSR #B262 EVAL BRKT ARGUMENT
DC47 BDB740 1101 JSR #B740 GET INTEGER IN X
DC4A #FE2 1102 CLR ,-5 FOR COUNTER
DC4C 1F10 1103 TFR I,D GET BAUD VALUE
DC4E 10032500 1104 CNPD #9600 HIGHEST LEGAL VALUE
DC72 1022FBC3 1105 LBHI FCERR ERROR IF HIGHER
DC76 4CE4 1106 CNTBD IMC ,S COUNT SUBTRACTION
DC78 03012C 1107 SUBD #300 DIVIDE BAUD BY 300
DC7B 1020F8BA 1108 LBHI FCERR NOT A VALID VALUE
DC7F 26F5 1109 BNE CNTBD CONTINUE SUBTRACTION
1110 * GOT A VALID MULTIPLE OF 300
DC81 3502 1111 PULS A GET RESULT
DC83 5F 1112 CLR# POWER COUNTER
DC84 BDC5E 1113 LDX #BDCNST POINT X TO BAUD CONSTANTS
DC87 44 1114 SFTAGN LSRA BIT INTO CARRY
DC88 2503 1115 BCS GETCON GOT BIT GET CONSTANT
DC8A 5C 1116 INCB COUNT SHIFT
DC8B 20FA 1117 BRA SFTAGN GO SHIFT AGAIN
DC8D A685 1118 GETCON LDA B,X GET BAUD RATE
DC8F 9796 1119 STA BAUDRT SET BAUD RATE
DC91 #F95 1120 CLR BDFLAG CLEAR TO ENABLE SERIAL PORT
1121 * AND SET LSB OF BAUD RATE
DC93 39 1122 *
1123 RTS ALL DONE
1124 *****
1125 * "LDIR" COMMAND, PRINT DIRECTORY
1126 *
DC94 C6FE 1127 LDIR LDB #2 POINT DEVNUM TO PRINTER
DC96 D76F 1128 STB DEVNUM
DC98 7ECBCF 1129 JHP #0016 DO DIR COMMAND
1130 *****
1131 IFDF PARPRT ASSEMBLE FOR PARALLEL PORT

```

dition. From a programmer's point of view, the PIA is addressed as follows:

```

FF24 Bit 0 - 6 unused
      Bit 7 printer busy line

FF25 Control port for above (set
      to $4)

FF26 Bit 0 - Parallel port bit 1
      Bit 1 - Parallel port bit 2
      Bit 2 - Parallel port bit 3
      Bit 3 - Parallel port bit 4
      Bit 4 - Parallel port bit 5
      Bit 5 - Parallel port bit 6
      Bit 6 - Parallel port bit 7
      Bit 7 - Parallel port bit 8

FF27 Bit 0 - 0
      Bit 1 - 0
      Bit 2 - 1

      Bit 3 - STROBE (BAR)
      Bit 4 - 1
      Bit 5 - 1
      Bit 6 - not used
      Bit 7 - BUSY FLAG (1 when

```

not busy)

This should provide the information you need to incorporate the parallel port into FLEX. Drop me a line if you have trouble.

Coming Attractions

One of the glaring omissions from BASIC is its ability to trap and deal with system errors in a graceful way. We will add this trapping, along with fully spelled out error messages, both on the screen as well as available in a string variable; plus variables identifying the type of error and the line number it occurred.

If you would like the entire DOS-PATCH program source, along with binary files with and without the parallel port driver for DECB 1.0 and DECB 1.1, just send me a disk (no cassettes please) along with \$6 and a stamped, addressed disk mailer. I will load the disk and return it to you promptly. Address this request or any questions to Colin Stearman, 143 Ash Street, Hopkinton, MA 01748.

```

1132 *
1133 * "PARALLEL" COMMAND CODE AND OUTPUT ROUTINE
1134 PARA LDD #01CA 120 BAUD DELAY
1135 * SET MB3 TO 1 FOR PARALLEL PORT
DC9E DD95 1136 STD BDFLAG TO MAKE PARALLEL ACTIVE
DCA0 39 1137 RTS
1138 *****
1139 * Parallel port output routine
1140 * This is called by the modified jump at #168
DCA1 #D95 1141 PAROUT TST BDFLAG IF NOT ZERO THEN PARALLEL
DCA3 1027EEA3 1142 LBDEQ #A015 DO SERIAL OUTPUT
DCA7 3402 1143 PSHS A SAVE VALUE
DCA9 966F 1144 LDA DEVNUM GOING TO DEVICE -2?
DCAB 81FE 1145 CNPA #2
DCAD 3502 1146 PULS A RECOVER CHAR, FLAGS DONT CHANGE
DCAF 1026EE97 1147 LBME #015 NOT DOING DEVICE #2
1148 *
1149 * PARALLEL OUTPUT WANTED
DCB3 0100 1150 CNPA #000 WAS IT A CR?
DCB5 2703 1151 BEQ #ASCRC
DCB7 #C9C 1152 INC <9C INCREMENT LINE PRINT POSITION
DCB9 0C 1153 FCB #0C SKIP NEXT 2 BYTES
DCBA #F9C 1154 #ASCRC CLR <9C LINE COUNTER
DCBC 3411 1155 PSHS CC,X PRESERVE BASIC VALUES
DCBE BEFF26 1156 LDX #DATA POINT X TO PIA
DCC1 #D1E 1157 CHKRDY TST -2,X BUSY IF LINE 7 HI
DCC3 20FC 1158 BHI CHKRDY WAIT UNTIL LOW
DCC5 A704 1159 STA ,X DATA REGISTER
DCC7 3511 1160 PULS CC,X RECOVER VALUES
DCC9 3262 1161 LEAS 2,S OLD RETURN OFF STACK
DCCB 39. 1162 RTS TO ORIGINAL CALLER
1163 *****
1164 ENDC
1165 OPT LIS
1166
1167
DCCD 1168 ZLAST EQU *-1 last used address value
1169 *
1170 * ZLAST must not be greater than $DFFF for
1171 * DOS 1.0 and $DEFF for DOS 1.1. The latter
1172 * has the OS-9 boot program and SWI set routines
1173 * from $DF00 to $DF4C
1174 *
1175 *
1184 OPT LIS
1185 END ADDCOM
D994 1185
NO ERROR(S) DETECTED

```

A Night Of Horrors In Lurkley Manor



It's Halloween. You're locked in Lurkley Manor with an assortment of bizarre people and other unearhtly things. You must use your skill and logic to escape the house without getting bumped off.

Lurkley Manor comes wrapped in a "fiendishly" long program listing which requires 32K and Extended Color BASIC. I hope you'll find the program worth the work, for it's an all-graphics logic puzzle with 16 scenes, 11 characters, animation and text in the Extended graphics mode. You will see the instructions when the program runs.

If your computer does not support the POKE 65495,0 "speed poke", you must delete it from lines 2 and 1830 of the listing.

If you wish the text to stay on the screen longer before erasing, in Line 2630 increase the timer loop from FOR T=1 to 1000 to FOR T=1 TO 2000 or greater. The title sequence and instructions may be skipped by inserting GOTO 160: at the start of Line 120. Do not renumber lines 1 to 64; this will cause the program to work incorrectly.

A self-addressed, stamped envelope gets you the step-by-step instructions for escaping Lurkley Manor. Send your request to 1493 Mt. View Ave., Chico, CA 95926.

15	127	1330	122
35	217	1450	110
62	18	1560	22
180	233	1710	237
280	247	1850	163
380	56	1980	235
500	190	2110	205
610	90	2220	39
710	121	2310	86
790	76	2400	121
880	144	2480	153
1020	88	2580	72
1110	115	END	45
1200	109		

The listing:

1 REM * LURKLEY MANOR * TRS-80 E
XTENDED COLOR BASIC / RAMELLA
2 CLEAR 500: POKE 65495,0

3 W\$="DIRECTION?": DI\$="NEWS": W
A\$="YOU RUN INTO A WALL.": GOTO
100

4 X=X-6: RETURN
5 Z\$="30353838": RETURN
6 Z\$="10135053": RETURN
7 Z\$="2029606903930797": RETURN
8 Z\$="700000040474747979094049":
RETURN

9 Z\$="111179792970": RETURN
10 Z\$="79000070700505090965": RE
TURN

11 Z\$="3033": RETURN
12 Z\$="402222272749": RETURN
13 Z\$="406262676749": RETURN
14 Z\$="43472424262664646666": RE
TURN

15 Z\$="42481575": RETURN
16 Z\$="2937": RETURN
17 Z\$="2575": RETURN
18 Z\$="4949": RETURN
19 Z\$="7009": RETURN
20 Z\$="60200108296978716228": RE
TURN

21 Z\$="234040482959": RETURN
22 Z\$="022020606073737570707090
979": RETURN
23 Z\$="0220206060737375753575787
86969292908": RETURN
24 Z\$="000606765059": RETURN
25 Z\$="700000040464647575777595
9191908": RETURN

26 Z\$="7020200202070729297979787
8757515": RETURN
27 Z\$="00707009": RETURN
28 Z\$="0110106060717173736464141
40303011405050808191969697878757
564": RETURN

29 Z\$="7414140303010110106060717
178786969191908": RETURN
30 Z\$="43434747": RETURN
31 Z\$="43432937": RETURN
32 Z\$="72050577": RETURN
33 Z\$="12721575": RETURN
34 Z\$="02757507": RETURN

35 Z\$="0220206060717173736464343
4373939": RETURN
36 Z\$="5634341414050507071818686
8777772725050111102": RETURN
37 Z\$="0920205050797414": RETURN
38 Z\$="0009096969787864640464737
37171606000": RETURN

39 Z\$="7160601010010108081919696
978": RETURN
40 Z\$="000909696975757272505000"
: RETURN
41 Z\$="7000000909790565": RETURN

42 Z\$="700000090565": RETURN
43 Z\$="6010100101070729296969667
646": RETURN
44 Z\$="000970790575": RETURN
45 Z\$="206029694049": RETURN
46 Z\$="70777759593939060604": RE
TURN

47 Z\$="000905702379": RETURN
48 Z\$="00090979": RETURN
49 Z\$="0900004545707079": RETURN
50 Z\$="090000797970": RETURN
51 Z\$="1070707969190801": RETURN
52 Z\$="0901107171746505": RETURN
53 Z\$="10707079691908017956": RE
TURN

54 Z\$="090000707073736464040479"
: RETURN
55 Z\$="70000004047474797909": RE
TURN
56 Z\$="00704049": RETURN
57 Z\$="000909797970": RETURN
58 Z\$="00494970": RETURN
59 Z\$="0029294545696970": RETURN
60 Z\$="00790970": RETURN

61 Z\$="004545704549": RETURN
62 Z\$="007070090979": RETURN
63 IF QS=CHR\$(32) THEN RETURN
64 FOR H=1 TO LEN(Z\$) STEP 4: LI
NE(X+VAL(MID\$(Z\$,H,1)),Y+VAL(MID
\$(Z\$,H+1,1)))-(X+VAL(MID\$(Z\$,H+2
,1)),Y+VAL(MID\$(Z\$,H+3,1))),PSET
: NEXT: RETURN

100 PPOKE 3,1: PCLS: SCREEN 1,1
110 PI\$="L10;01;G;F;D#;D": P2\$="L10;G;G#;G;F;D#": P3\$="L10;F;L4;G": P4\$="L4;D;L1;C": COLOR 3,0
120 GOSUB 1740: GOSUB 2650: KZ=1
: GOSUB 1600

130 A\$="I AM, OF COURSE, IGOR. Y
OU ARE AN UNLUCKY FOOL NOW LOCKE
D IN LURKLEY MANOR.": GOSUB 1530

140 A\$="TO TRAVEL, PRESS N FOR N
ORTH, E FOR EAST, W FOR WEST, S
FOR SOUTH.": GOSUB 1530
150 A\$="IN EVERY SCENE YOU FACE
NORTH.": GOSUB 1530: A\$="FIND TH
E ATTIC TO ESCAPE... PERHAPS.":
GOSUB 1530
160 GOSUB 2650: A\$="THE GREAT HA
LL": GOSUB 1530: GOSUB 2510: GUS
UB 2520: GOSUB 2530

```

170 DRAW "BH100,95;U50L10U5R70D5
L10050L50": CIRCLE(125,76),21,,1
.5,.4,.12: T1=90: B1=T1: PAINT(1
25,76),3,3
180 IF PV=0 THEN AS="IT IS EMPTY
EXCEPT FOR A ROARING FIREPLACE.
": GOSUB 1530: PV=1
190 AS=W$: GOSUB 1530
200 FOR J1=3 TO 4: COLOR J1,0: L
INE(112+RND(25),93)-(112+RND(26)
,60-RND(5)),PSET: NEXT
210 X$=INKEY$: IF X$="" OR INSTR
(D1$,X$)=0 THEN 200
220 IF BU$="OK" AND X$="N" THEN
BU$="": PA$="PA": AS="YOU DOUSE
THE FIRE AND ENTER": GOSUB 1530:
PAINT(125,56),2,1: GOTO 2590
230 IF X$="N" THEN 1310 ELSE IF
X$="E" THEN 240 ELSE IF X$="W" T
HEN 880 ELSE AS="THE DOOR BEHIND
YOU IS LOCKED. REMEMBER?": GOSU
B 1530: GOTO 200
240 GOSUB 2650: AS="SCULLERY": G
OSUB 1530: GOSUB 2510: GOSUB 252
0: GOSUB 2530: DS=180: GOSUB 254
0
250 FOR V=32 TO 168 STEP 2: DRAW
"C2;IM"+STR$(V)+",105;"+U30;C3
;E10": NEXT V: FOR V=74 TO 104 S
TEP 2: DRAW"BM170,"+STR$(V)+";E1
0": NEXT V
260 FOR V=50 TO 75 STEP 25: CIRC
LE(V,70),10,1,.6: PAINT(V,70),1,
1: CIRCLE(V+4,65),6,,.2,.6,1: NEX
T
270 FOR V=140 TO 160 STEP 20: CI
RCLE(V,70),7,4,.6: PAINT(V,70),4
,4: NEXT: CIRCLE(153,65),20,3,.9
,.5,1: FOR V=50 TO 20 STEP -1: C
IRCLE(153,V),4,3,.7,.5,1: NEXT
280 CIRCLE(128,100),30,4,.3: PAI
NT(128,100),4,4: DRAW"C4;BH105,1
25;U20E9D25": DRAW"BM145,125;U20
E9D25"
290 IF QA=0 THEN QA=1: AS="A DOU
BLE SINK, AN OLD COOK STOVE AND
A BARE TABLE.": GOSUB 1530
300 AS=W$: GOSUB 1530: GOSUB 266
0: IF HY=1 THEN 1340 ELSE IF HY=
2 THEN 950 ELSE IF HY=3 THEN 160
ELSE AS=W$: GOSUB 1530: GOTO 3
00
310 GOSUB 2650: AS="FLAGON ROOM"
: GOSUB 1530
320 CIRCLE(128,60),30,,.2,5: DRAW
"BM99,60;H25U35R110D35G25": PAIN
T(97,55),2,3: PAINT(160,55),2,3
330 FOR V=115 TO 142 STEP 27: CI
RCLE(V,30),10,,.6: CIRCLE(V,30),
2,2: NEXT V
340 LINE(128,30)-(120,80),PSET:
LINE-(135,70),PSET: DRAW"BM120,9
0;G5H5R37G5H5": DRAW"C4;BH146,91
;D5G3F3"
350 JH$="D60L30U60R30F20G20": IF
F1=0 THEN DRAW"C4;BH60,74"+JH$:
PAINT(55,80),4,4
360 IF F2=0 THEN DRAW"C3;BH210,7
4;"+JH$: PAINT(190,80),3,3

```

```

370 IF PY=0 THEN AS="GOOD EFFININ
K! I AM COUNT DRACU- NEVER MIND
MY NAME!": GOSUB 1530: PY=1
380 IF F1=1 AND F2=1 THEN AS="I'
M OUT OF FLAGONS. GO AWAY.": GOS
UB 1530: GOTO 470
390 AS="WANT A FLAGON OF POIS- I
MEAN - DRINK AS YOU SEARCH? <Y>
ES <N>": GOSUB 1530
400 X$=INKEY$: IF X$<>"Y" AND X$
<>"N" THEN 400
410 IF X$="H" THEN AS="GO AWAY!"
: GOSUB 1530: GOTO 470
420 AS="WHICH COLOR - <O>RANGE O
R <D>LUE?": GOSUB 1530
430 X$=INKEY$: IF X$<>"O" AND X$
<>"B" THEN 430
440 IF X$="O" AND F1=0 THEN F1=1
: FL$="ORANGE": LINE(25,73)-(85,
134),PRESET,BF: AS="YOU GOT IT.
NOW LEAVE.": GOSUB 1530: GOTO 47
0
450 IF X$="B" AND F2=0 THEN F2=1
: FL$="BLUE": LINE(170,73)-(230,
134),PRESET,BF: AS="YOU'RE WELCO
ME I'H SURE. NOW RUN BEFORE THE
FULL MOON RISES!": GOSUB 1530: G
OTO 470
460 IF X$="B" AND F2=1 THEN AS="
SURELY IT'S OBVIOUS THAT'S A FRU
ITLESS GUESS. DEPART LEST I BITE
YOU!": GOSUB 1530: GOTO 470
470 AS=W$: GOSUB 1530: GOSUB 266
0: IF HY=1 THEN 880 ELSE AS=W$:
GOSUB 1530: GOTO 470
480 GOSUB 2650: AS="DINING ROOM"
: GOSUB 1530: GOSUB 2510: GOSUB
2520: DS=170: GOSUB 2540
490 DRAW"BM236,60;D31H6U25E6D5F6
G6": GOSUB 2670
500 DRAW "C2;BH60,120;U40L5U5R10
5D5L5D40L5U35L85D35L5": PAINT(62
,118),2,2
510 FOR V=60 TO 160 STEP 30: CIR
CLE(V,71),8,4,.4,1,.5: FOR U=1 T
O 4: CIRCLE(V+RND(2)-1,70),RND(5
),RND(3)+1: NEXT U,V
520 IF QZ=1 THEN QZ=0: GOTO 560
ELSE AS="THAT FOOD LOOKS GOOD. W
ANT A SNACK? <Y>ES OR <N>O.": GO
SUB 1530
530 X$=INKEY$: IF X$<>"Y" AND X$
<>"N" THEN 530 ELSE IF X$="N" TH
EN 560
540 XZ=1: FOR V=50 TO 150 STEP 3
0: AS=HLD$( "SLURPSNORFSCHMFGASP"
,XZ,5)+": GOSUB 1530: LINE(V,6
0)-(V+18,74),PRESET,BF: PLAY "T1
28;05;CDECD": XZ=XZ+5: NEXT
550 AS="UH-OH! ANOTHER PIGGY BEC
OMES A DINING ROOM VICTIM. THE E
ND!": KZ=1: GOSUB 1530: GOTO 261
0
560 AS=W$: GOSUB 1530: GOSUB 266
0: IF HY=3 THEN GL=2
570 ON HY GOTO 580,1140,1440,880
580 GOSUB 2650: L=0: FOR V=135 T
O 40 STEP -10
590 DRAW "BH"+STR$(L)+", "+STR$(V

```

```

)+U5R20D5L20U5E5R20G5"
600 DRAW "BH"+STR$(230-L)+", "+ST
R$(V)+U5R20D5L20U5H5R20F5"
610 PAINT(234-L,V-2),3,3: PAINT(
L+2,V-2),4,3: L=L+5: NEXT
620 LINE(60,35)-(180,135),PSET,B
: PAINT(65,133),4,3: PAINT(182,1
33),3,3
630 FOR V=80 TO 153 STEP 31: LIN
E(V,5)-(V+25,35),PSET,B: NEXT
640 IF LK=0 THEN LK=1: AS="YOU A
RE AT TWO STAIRWAYS TOPPED BY TH
REE DOORS.": GOSUB 1530
650 AS="CHOOSE <L>EFT STAIR, <R>
IGHT STAIR OR <S>OUTH.": GOSUB 1
530
660 X$=INKEY$: IF X$="" OR INSTR
("LRS",X$)=0 THEN 660
670 IF X$="S" THEN 480
680 IF FL$="ORANGE" AND X$="L" O
R FL$="BLUE" AND X$="R" THEN GOS
UB 2630: GOTO 720
690 AS="AS YOU NEAR THE TOP, THE
ENTIRE ARRANGEMENT COLLAPSES.":
GOSUB 1530
700 L=8: FOR K=1 TO 8: FOR V=36
TO 134 STEP L: LINE(2,V)-(253,V)
,PRESET: NEXT V: L=L-1: NEXT K
710 AS="IN YOUR NEXT LIFE, REMEN
BER THE FLAGON AND STAIR COLORS
MUST MATCH.": GOSUB 1530: GOSUB
1820: GOTO 710
720 AS="YOU ASCEND TO THE DOORS.
OPEN <L>EFT, <M>IDDLE OR <R>IGHT
>. WHICH?": GOSUB 1530
730 X$=INKEY$: IF X$="" OR INSTR
("LMR",X$)=0 THEN 730 ELSE IF X$
="H" THEN 2130
740 GOSUB 2650: FOR V=0 TO 255 S
TEP 85: LINE(V,0)-(V,135),PSET:
NEXT
750 AS="YOU SLOWLY OPEN THE DOOR
.": GOSUB 1530
760 IF X$="L" THEN GOSUB 1960 EL
SE GOSUB 1840
770 IF X$="L" AND FL$="ORANGE" O
R X$="R" AND FL$="BLUE" OR YO=0
AND X$="R" OR YO=1 AND X$="L" TH
EN 2200
780 IF X$="L" THEN YO=1: FL$=FL$
+"X": AS="A FORBODING BUT FRIEND
LY MUMMY APPEARS. IT GESTURES OV
ER THE BLUE FLAGON.": GOSUB 1530
: AS="ITS ADVICE: SHARE THE POTA
BLE WITH SOMEONE WHOSE NAME BEGI
NS AS ITS COLOR.": GOSUB 1530: G
OTO 800
790 YO=YO+1: FL$=FL$+"X": AS="A
RATTLING SKELETON IS THERE.": GO
SUB 1530: AS="IT TELLS YOU: SAY
YES TO THE LADY.": GOSUB 1530
800 AS="CONSIDERABLY WISER, YOU
DESCEND THE STAIRS WITH YOUR FLA
GON.": GOSUB 1530: GOTO 580
810 GOSUB 2650: AS="YOU GET A WH
IFF OF HAY.": GOSUB 1530
820 CIRCLE(128,65),70,,.8,.6,.1:
CIRCLE(128,65),70,,.8,.2,.3: CI
RCLE(128,65),70,,.8,.4,.5

```

```

830 DRAW "BM185,95;D30L22H9C1L46
C3G9L22U25": CIRCLE(205,58),12,,
.7,0,.4
840 CIRCLE(55,45),20,,1,.3,.9: F
OR V=19 TO 29 STEP 10: CIRCLE(V,
65),30,,1,0,.25: NEXT: CIRCLE(48
,45),3
850 AS="AN OBESE, DIM-WITTED ELE
PHANT IS THERE.": GOSUB 1530: AS
="HAPPY TO SEE YOU, IT RUSHES FO
RWARD IN GREETING...": GOSUB 153
0
860 IF BU$="BU" THEN BU$="OK": A
S="AND FILLS THE BUCKET WITH WAT
ER FROM ITS TRUNK.": GOSUB 1530:
AS="IT TRUMPETS: GO DOUSE SOMET
HING.": GOSUB 1530: GOTO 1340
870 PLAY "L4;O1;C;O4;L1;C": AS="
AND STEPS ON YOU. YOU FEEL MIGHT
Y LOW AS EVERYTHING ENDS.": KZ=1
: GOSUB 1530: GOTO 2610
880 GOSUB 2650: AS="DRAWING ROOM
": GOSUB 1530
890 GOSUB 2510: GOSUB 2520: DS=4
6: GOSUB 2540: GOSUB 2670
900 IF YO<3 THEN GOSUB 1680
910 X=2: FOR V=80 TO 150 STEP 2:
DRAW "C"+STR$(X)+";BH"+STR$(V)+
",120;"+"U2OE2R2E2R2E2R2U2OE2R2E
2C2D4OC3C5L3G5L3G5": X=X+1: IF X
=5 THEN X=2: NEXT ELSE NEXT
920 IF YO<3 THEN GOSUB 2270
930 AS=W$: GOSUB 1530: GOSUB 266
0: IF HY=3 THEN GL=1
940 ON HY GOTO 480,160,1440,310
950 GOSUB 2650: GH=0: AS="THE PA
RAPEE": GOSUB 1530
960 FOR I=64 TO 129 STEP 5: N1=N
I+1: IF N1/2=INT(N1/2) THEN V=1
ELSE V=5
970 FOR G=V TO V+200+RND(20) STE
P 10: LINE(G,I)-(G+10,I+5),PSET,
B: NEXT G,I
980 FOR V=2 TO 80 STEP 2: CIRCLE
(V,63),20,4,1,.5,.75: CIRCLE(V,6
3),12,3,1.8,.77,1: NEXT
990 IF JL=0 THEN AS="AN ARCHED W
ALKWAY OPENS ON A CRUMBLING BRIC
K WALL.": GOSUB 1530: JL=2
1000 AS="IT'S HUNDREDS OF FEET D
OWN -- SOUTH, EAST AND NORTH.":
GOSUB 1530: AS=W$: GOSUB 1530: K
Z=1: GOSUB 2660
1010 KZ=0: IF HY=3 THEN 240
1020 IF GH=1 AND PA$="PA" THEN 2
550
1030 IF GH=1 THEN CIRCLE(220,33)
,10,1,.5,0,.5: CIRCLE(220,40),10
,2,.5,.5,0: AS="YOU JUMP.": GOSU
B 1530: AS="YOU FOOL!": GOSUB 15
30: FOR VB=240 TO 1 STEP -5: SOU
ND VB,1: NEXT VB: AS="THE END":
KZ=1: GOSUB 1530: GOTO 2610
1040 CIRCLE(220,60),26,2,2,.5,1:
DRAW"C2;BH200,60;E5F5E5F5E5F5E5
F5"
1050 CIRCLE(220,33),10,,.5,0,.5:
CIRCLE(213,25),5: CIRCLE(228,25
),5
1060 GH=1: AS="A GHOST APPEARS I
N MID-AIR, SAYING... RECONSIDER!
GO WEST OR BE AS I!": GOSUB 153
0: GOTO 1000
1070 GOSUB 2650: AS="LABORATORY"
: GOSUB 1530: GOSUB 2510: GOSUB
2530
1080 FOR V=40 TO 80 STEP 10: FOR
V1=100 TO 140 STEP 10: LINE(V1,
V)-(V1+10,V+10),PSET,B: NEXT V1,
V: DRAW"C1;BH160,40;R20": DRAW"BM
50,95;R40"
1090 COLOR 2,1: LINE(170,120)-(2
15,134),PSET,BF: LINE(175,70)-(2
10,119),PSET,B: COLOR 3,1: GOSUB
1890: DRAW"C1;BH176,95;R33"
1100 DRAW"C2;BH190,90;E10H10E10H
10E10H10L63D9"
1110 AS="PROFESSOR FUDDLES PERFO
RMS UNSPEAKABLE EXPERIMENTS ON S
MALL HELPLESS ANOEBAS.": GOSUB 1
530
1120 FOR V=1 TO 100: PAINT(101+R
ND(42),43+RND(43)),RND(2)*2,3: P
SET(176+RND(34),71+RND(48),RND(2
)*2): SOUND RND(240),1: NEXT
1130 AS=W$: GOSUB 1530: GOSUB 26
60: IF HY=2 AND CV=1 THEN CV=0:
GOSUB 2100: GOTO 1070 ELSE IF HY
=3 THEN 1340 ELSE AS=W$: GOSUB
1530: AS="THIS GIVES YOU A CHANC
E TO WATCH SOME MORE AS...": GOS
UB 1530: GOTO 1110
1140 GOSUB 2650: AS="FLAGON SPIL
LING ROOM": GOSUB 1530
1150 GOSUB 2510: FOR V=1 TO 100:
PSET(RND(253)+2,RND(133)+2,2):
NEXT V
1160 AS="THERE ARE STRANGE SMALL
THINGS ALL AROUND.": GOSUB 1530
1170 IF FL$<>"ORANGE" AND FL$<>"
BLUE" THEN 1290
1180 AS="THIS IS THE FLAGON SPIL
LING ROOM. YOU HAVE A FLAGON. YO
U SPILL IT.": GOSUB 1530: AS="TH
E FUMES REANIMATE HUMMIFIED SPID
ERS WHICH GROW HUGE AND ATTACK."
: KZ=1: GOSUB 1530: PCLS 1
1190 DIM A(1,25),B(7): PCLS 1: C
IRCLE(15,14),3,,.5: CIRCLE(15,10
),13,,.8
1200 FOR V=8 TO 23 STEP 5: V$=ST
R$(V)+",17;": IF V<16 THEN J$="
G5F5" ELSE J$="F5G5"
1210 DRAW "BM"+V$+J$: NEXT V: PS
ET(11,7,4): PSET(18,7,4)
1220 GET(2,2)-(33,33),A,G: PCLS
1
1230 X=RND(7)-1: P=X*36+8: FOR J
=1 TO RND(25): B(X)=B(X)+1
1240 PUT(P,B(X))-(P+31,B(X)+25),
A
1250 IF B(X)>110 THEN 1270
1260 NEXT J: GOTO 1230
1270 AS="OH WHAT A SAD ENDING!":
GOSUB 1530: PAINT(0,0),4,3: COL
OR 1,1: AS="E-E-E-E-E-E-E-K!": KZ=1:
X=64: Y=160: GOSUB 1540
1280 PLAY "T1": GOSUB 1820: GOTO
1280
1290 AS="YOU'RE VERY LUCKY . . .
FOR SOME REASON": GOSUB 1530
1300 GOSUB 2650: AS="DIRECTION?":
GOSUB 1530: GOSUB 2660: IF HY=
3 THEN 480 ELSE AS=W$: GOSUB 15
30: GOSUB 2660: GOTO 1140
1310 AS="YOU WALK DIRECTLY INTO
THE FIREPLACE.": GOSUB 1530
1320 AS="YOU MUST THINK THIS TES
T IS A CINCH. FOR YOU IT'S A SIN
GE. T-H-E E-N-D": GOSUB 1530: GO
TO 2610
1330 AS="THE DOOR BEHIND YOU IS
LOCKED. REMEMBER?": GOSUB 1530:
GOTO 160
1340 GOSUB 2650: AS="GYMNASIUM":
GOSUB 1530: GOSUB 2510: GOSUB 2
520
1350 PAINT(228,101),3,3: DS=100:
GOSUB 2540: GOSUB 2030: IF YO=3
THEN GOSUB 2490: GOTO 1400
1360 IF BU$="BU" AND SD=0 THEN S
D=1: AS="BLURTON SAYS: GO SEE TH
E ELEPHANT.": GOSUB 1530: GOTO 1
400
1370 IF VX=0 THEN VX=1: AS="SMIR
KING BUT SILENT, BLURTON SHARPE
PRACTICES HIS PISTOL SKILL.": GO
SUB 1530
1380 GOSUB 2490
1390 ES=LEFT$(FL$,1): IF ES="B"
OR ES="O" THEN GOSUB 2440
1400 AS="DIRECTION?": GOSUB 1530
: GOSUB 2660: IF PI$="" AND HY=1
OR PI$="" AND HY=3 THEN AS="YOU
ENTER SHARPE'S LINE OF FIRE, AN
D HE PLUGS YOU. THE END.": GOSUB
1530: GOTO 2610
1410 IF PI$="EMPTY" AND HY=1 THE
N 810
1420 IF PI$="EMPTY" AND HY=3 THE
N AS=W$: GOSUB 1530: GOTO 1400
1430 IF HY=4 THEN 240 ELSE IF HY
=2 THEN 1070
1440 GOSUB 2650: FOR V=1 TO 79 S
TEP 3: CIRCLE(128,55),V,RND(2)+2
: NEXT V
1450 AS="THIS MEANS JUMPING THRO
UGH A BEAUTIFUL STAINED GLASS WI
NDOW. <Y>ES OR <N>O": GOSUB 1530
1460 X$=INKEY$: IF X$<>"Y" AND X
$<>"N" THEN 1460 ELSE IF X$="N"
AND GL=2 THEN 560 ELSE IF X$="N"
AND GL=1 THEN 930
1470 DRAW"C2;BH128,40;U30G10D20H
30D10F20L30F20G30R5U5R13U5E20F40
U10R40H40U30G20": PAINT(128,65),
1,2
1480 FOR X=255 TO 1 STEP 10: SOU
ND X,1: NEXT: AS="YOU JUMP": GOS
UB 1530
1490 IF PA$="PA" THEN AS="YOUR P
ARACHUTE RIPS ON JAGGED GLASS AN
D IS USELESS.": GOSUB 1530
1500 AS="IT'S THE END. YOU FALL
. . .": GOSUB 1530

```



```

1510 A$="... AND FALL ...": GOSUB
B 1530: GOSUB 1820: GOTO 1510
1520 IF FI=2 THEN DRAW"C3;BM200,
74"+JH$: PAINT(180,80),3,3
1530 COLOR 3,1: X=1: Y=140: A$=A
$+CHR$(32)
1540 FOR P=1 TO LEN(A$): Q$=MID$(
A$,P,1)
1550 K=ASC(Q$)-31: IF K<1 OR K>5
9 THEN 1590
1560 ON K GOSUB 4,5,6,7,8,9,10,1
1,12,13,14,15,16,17,18,19,20,21,
22,23,24,25,26,27,28,29,30,31,32
,33,34,35,36,37,38,39,40,41,42,4
3,44,45,46,47,48,49,50,51,52,53,
54,55,56,57,58,59,60,61,62
1570 GOSUB 63: X=X+12: IF Q$=CHR
$(32) AND INSTR(MID$(A$,P+1,INT(
(255-X)/12)),CHR$(32))=0 THEN X=
1: Y=Y+14
1580 IF Y>182 THEN GOSUB 2630
1590 NEXT P: GOSUB 2630: RETURN
1600 CIRCLE(128,90),55,,1.5,.6,.
09
1610 FOR V=120 TO 128: FOR U=110
TO 170 STEP 60: CIRCLE(U,V),15,
,2: NEXT U,V
1620 CIRCLE(155,75),13,,2.5,.5,1
: DRAW"BM137,75;R30": CIRCLE(157
,75),13,,1,0,.5: CIRCLE(85,97),1
1,,1,0,.5
1630 DRAW "BM155,120;H15G15L20H8
U15L20U22"
1640 CIRCLE(100,62),30,,.8
1650 FOR V=90 TO 110 STEP 20: CI
RCLE(V,55),9,,.7: CIRCLE(V,58),3
: NEXT
1660 DRAW "BM101,55;D3G7D5F3R2E5
U2"
1670 LINE (97,80)-(107,77),PSET:
RETURN
1680 COLOR 2,1: CIRCLE(200,130),
20,,3,.5,0: LINE(180,130)-(220,1
30),PSET
1690 DRAW "BM195,75;U6L1H3U6E3R3
F4E4R2F3D6G3D6G3"
1700 LINE(200,120)-(200,70),PRES
ET: PAINT(200,112),2,2
1710 CIRCLE(200,40),9,,1.6: CIRC
LE(200,66),16,,1.1,.35,.16: PAIN
T(200,40),1,2
1720 PSET(198,37,3): PSET(202,37
,3): CIRCLE(200,41),1: CIRCLE(20
0,44),2,,.7,0,.5
1730 COLOR 4,0: FOR V=10 TO 14:
CIRCLE(200,46),V,,1.7,.5,1: NEXT
: COLOR 3,0: RETURN
1740 PHODE 4,1: COLOR 1,0: PCLS0
: SCREEN 3,2
1750 DRAW"BM10,134;E10R3U2E5R40U3
R50U3R150"
1760 LINE(116,110)-(90,30),PSET
1770 DRAW"BM190,30;L10U10R40D10L1
0D10R40U10L5U5R15D5L5D10R30U20L1
0U8R30D8L10D20R40"
1780 LINE-(220,111),PSET
1790 FOR V=1 TO 12: CIRCLE(180,1
11),V,,2,.5,1: NEXT
1800 FOR V=1 TO 10: CIRCLE(30,20

```

```

),V: NEXT
1810 KZ=1: A$=STRING$(9,32)+"LUR
KLEY MANOR"+STRING$(11,32)+"BY R
ICHARD RAHELLEA": GOSUB 1530: KZ=
0
1820 POKE 65494,0: POKE 223,6: F
OR G=1 TO 2: FOR T=1 TO 2: PLAY
P1$: NEXT T: PLAY P2$: IF G=1 TH
EN PLAY P3$ ELSE PLAY P4$
1830 NEXT G: POKE 65495,0: RETUR
N
1840 PAINT(220,5),3,3: COLOR 1,3
: CIRCLE(210,16),11,,1.5,.45,.05
6: CIRCLE(210,19),9,,2.2,0,.5
1850 PSET(208,11,1): PSET(213,11
,1): DRAW "BM210,16;D3": DRAW "B
M209,24;R4"
1860 DRAW "BM190,69;U10R3U17E5R2
5E15U3H10": DRAW "BM210,34;D40"
1870 V1=14: V2=.7: V3=47: FOR V=
1 TO 5: CIRCLE(210,V3),V1,,V2,.4
5,.05: V1=V1-2.5: V2=V2-.1: V3=V
3+.5: NEXT V
1880 FOR V=67 TO 70: CIRCLE(210,
V),14,,.6,0,.5: NEXT: DRAW"BM188
,132;E7U10H3U30E7R20F20G20F7": R
ETURN
1890 CIRCLE(70,90),25,,1.5,.45,.
99: CIRCLE(80,90),25,,1.5,.5,.6:
CIRCLE(53,95),7
1900 DRAW"BM48,103;D15G5R10E2R5E
10"
1910 CIRCLE(68,85),18,,1.6,.75,.
25
1920 CIRCLE(70,47),15,,1.2: DRAW
"BM80,51;L4H2": CIRCLE(76,41),3
,,.7
1930 FOR V=5 TO 16: CIRCLE(72,47
),V,,1.2,.5,.7: NEXT V
1940 DRAW "BM63,120;D14R17E2U4H2
L5U5E2U5E7U2E7U3E2U1E2U7L9": PAI
NT(65,132),3,3
1950 PAINT(70,100),4,3: RETURN
1960 FOR V=21 TO 145 STEP 1: LIN
E(6,V+(RND(10)-20))-(75,V+RND(10
)-20),PSET: NEXT V: COLOR 3,0
1970 COLOR 2,0: DRAW "BM35,80;D5
0L20U10R10U80G10D30L10U35E20R5"
1980 CIRCLE(39,17),14,,1.4,.12
1990 DRAW "BM35,80;R10D50R20U10L
10U80F10D30R10U35H20L5"
2000 LINE(0,0)-(85,135),PSET,B:
PAINT(5,5),4,2: PAINT(5,5),1,2
2010 COLOR 3,0: LINE(0,0)-(85,13
5),PSET,B: PAINT(5,5),1,3
2020 RETURN
2030 COLOR 2,1
2040 DRAW"BM202,130;L13E4R5U75G1
2H15R7F7E11R20F15G15U7E7H9D75G8L
3U4E6R4L13R7U40": DRAW "BM199,70
;R15"
2050 PAINT (202,68),2,2: PAINT (
202,72),4,2
2060 CIRCLE(206,33),10,,1.3: PAI
NT(206,33),0,2: PAINT(206,33),1,
3: COLOR 2,0: FOR V=1 TO 10: CIR
CLE(206,28),V,,.8,.5,1:NEXT V: C
OLOR 3,0
2070 DRAW "BM203,37;F2R3E2": DRA

```

```

W"BM207,36;U4": PSET(202,30,3):
PSET(209,30,3)
2080 DRAW"BM173,47;U1L10U1R13E2G
3D7"
2090 DRAW"BM10,125;U5E5U50L5D15L
3U20R10U10R10D10R10D20L5U15L5D38
F3G10U25": RETURN
2100 GOSUB 2650: BU$="BU": A$="D
UNGEON": GOSUB 1530: GOSUB 2510:
A$="IT'S EMPTY EXCEPT FOR...":
GOSUB 1530
2110 DRAW"C2;BM195,103;R20D30L20
U30": PAINT(197,105),2,2: CIRCLE
(205,103),10,2,.5,1: A$="A NICE
LITTLE BUCKET": GOSUB 1530
2120 YO=YO+1: A$="YOU TAKE THE B
UCKET AND CRAWL BACK UP TO...":
GOSUB 1530: RETURN
2130 COLOR 3,1: L=1: FOR V=35 TO
59: FOR K=3 TO 1 STEP -2
2140 COLOR K,1: LINE(111+L,V)-(1
37,35),PSET: NEXT K: L=L+1: NEXT
V: COLOR 3,1
2150 A$="OH MY GOODNESS! A TRAPD
OOR OPENS AND DROPS YOU THROUGH.
": GOSUB 1530: A$="BUT DON'T WOR
RY TOO MUCH.": GOSUB 1530
2160 CIRCLE(136,96),40,2,1,.77,.
47: X$="C2;BM136,96": DRAW X$+"G
5H5G5H5G5H5G5": DRAW X$+"H5E5H5E
5H6E6H6R6"
2170 CIRCLE(150,75),10,2,.5: PAI
NT(128,133),2,2: CIRCLE(150,75),
2,4,.7
2180 A$="A RENEGADE FROM SOME GE
NTLER VIDEO PASTIME CATCHES YOU
BEFORE YOU HIT THE FLOOR.": GOSU
B 1530: A$="IT HAS EATEN RECENTL
Y BUT MANAGES TO PAC YOU IN. THE
END": GOSUB 1530
2190 GOSUB 1820: GOTO 2190
2200 IF X$="L" THEN V=2 ELSE V=1
82
2210 IF X$="L" THEN A$="A MUMMY
APPEARS, SAYING . . .": GOSUB 15
30: A$="I AM ENRAGED BY THE ORAN
GE OF YOUR FLAGON!": GOSUB 1530
2220 IF X$="R" THEN A$="THERE'S
A SKELETON IN THE CLOSET. IT SAY
S . . .": GOSUB 1530: A$="YOU CA
ME HERE TOO SOON, YET YOU ARE NO
W... THE LATE.": GOSUB 1530
2230 A$="YOU TAKE A PERMANENT PL
ACE ON THE OTHER SIDE OF THE MID
DLE DOOR.": GOSUB 1530
2240 CIRCLE(128,95),38,,2,.5,1:
X=107: Y=70: A$="R I P ": GOSUB
1540
2250 COLOR 2,1: FOR X=90 TO 166:
LINE(X,100)-(X+(3-RND(6)),100-R
ND(15)),PSET: NEXT X: COLOR 3,1:
2260 A$="IT LEADS TO A SECLUDED
GLEN JUST BEHIND LURKLEY MANOR.
YOUR FINAL STOP.": KZ=1: GOSUB 1
530: GOTO 2610
2270 A$="NOIRA DARK ASKS IF YOU
HAVE SOMETHING TO DRINK. <Y>YES O
R <N>NO.": GOSUB 1530

```

```

2280 X$=INKEY$: IF X$<>"Y" AND X
$<>"N" THEN 2280
2290 IF X$="N" AND FL$="" THEN A
$="DON'T FORGET ME.": GOSUB 1530
: RETURN
2300 IF X$="N" AND FL$<>" " OR X$
="Y" AND FL$="" THEN A$="SHE CAL
LS YOU A LIAR, AND RIGHTLY SO. S
HE FLIES INTO A RAGE.": GOSUB 15
30: GOTO 2370
2310 A$="SHE ASKS FOR A DRINK: <
Y>ES OR <N>O.": GOSUB 1530
2320 X$=INKEY$: IF X$<>"Y" AND X
$<>"N" THEN 2320
2330 IF X$="Y" AND LEFT$(FL$,2)=
"BL" THEN A$="IT'S BLUE! I HATE
PRUNE JUICE, OR WHATEVER IT IS!"
: GOSUB 1530: GOTO 2370
2340 IF X$="Y" AND FL$="ORANGE"
THEN A$="SHE SNIFFS AT IT, SAYIN
G: THE SKELETON MUST OKAY THIS F
IRST.": GOSUB 1530: RETURN
2350 IF X$="Y" AND FL$="ORANGEX"
THEN FL$="": YO=3: CV=1: A$="TH
AT TASTED VERY GOOD! MY ADVICE T
O YOU GO WALK THROUGH A WALL.":
GOSUB 1530: RETURN
2360 IF FD=0 AND X$="N" THEN FD=
1: A$="SHE VOWS: YOU'LL SLAKE MY
THIRST BEFORE YOU ESCAPE! SCRAM
!": GOSUB 1530: RETURN ELSE IF X
$="N" THEN A$="REMEMBER MY VOW."
: GOSUB 1530: RETURN
2370 COLOR 2,1: CIRCLE(200,66),1
6,1,1.1,.35,.67.
2380 DRAW"BM190,52;1.15H9L7F14R15
F2": CIRCLE(163,40),8,,.7: PAINT
(163,40),1,2: DRAW"BM163,10;D25"

2390 A$="SHE PRODUCES A BOMB.":
GOSUB 1530
2400 FOR X=10 TO 35: FOR V=4 TO
1 STEP -1: COLOR V,1: CIRCLE(163
,X),2,V: NEXT V: PLAY "T254;01;C
": NEXT X
2410 COLOR 4,1: FOR X=1 TO 60: F
OR V=0 TO 1: SCREEN 1,V: LINE(16
3,40)-(RND(255),RND(134)),PSET:

```

```

NEXT V,X: COLOR 3,1
2420 A$="NO ONE SURVIVES. WILL T
HIS TEACH YOU NOT TO FIB?": KZ=1
: GOSUB 1530: PLAY "T1"
2430 GOSUB 1820: GOTO 2430
2440 A$="BLURTON SAYS: I SEE YOU
HAVE A FLAGON. MAY I HAVE A SIP
- <Y>ES OR <N>O?": GOSUB 1530
2450 X$=INKEY$: IF X$<>"N" AND X
$<>"Y" THEN 2450 ELSE IF X$="Y"
AND FL$="BLUE" THEN A$="SEE MY H
OMMY - I MEAN MY MUMMY - TO GET
THIS APPROVED BEFORE I DRINK IT.
": GOSUB 1530: RETURN
2460 IF X$="Y" AND LEFT$(FL$,2)=
"OR" THEN A$="IT'S ORANGE STUFF,
WHICH I HATE!": GOSUB 1530: A$=
"HE FIRES. YOU FALL. IT ENDS.":
GOSUB 1530: GOTO 2610
2470 IF X$="N" THEN A$="BLURTON
SCOWLS: NO ONE REFUSES ME. I'M A
BLUEBLOOD!": GOSUB 1530: A$="HE
AIMS THE PISTOL AT YOU AND FIRE
S! THE END.": GOSUB 1530: GOSUB
2490: GOTO 2610
2480 A$="BLURTON DRAINS THE FLAG
ON AND KEEPS IT.": GOSUB 1530: A
$="HE SAYS: I LIKE IT. I PROMISE
NEVER TO SHOOT YOU.": GOSUB 153
0: FL$="": PI$="EMPTY": RETURN
2490 FOR V=1 TO 10: CIRCLE(159,4
5),3,4: POKE 65494,0: PLAY"T32;0
5;CB"
2500 CIRCLE(159,45),3,1: CIRCLE(
8+RND(20),60+RND(20)),3,4: NEXT:
POKE 65495,0: RETURN
2510 LINE(40,40)-(215,95),PSET,B
: LINE(40,40)-(0,0),PSET: LINE(2
15,40)-(255,0),PSET: LINE(215,95
)-(255,135),PSET: LINE(40,95)-(0
,135),PSET: RETURN
2520 DRAW"BM225,105;U50E20D90":
PAINT(228,101),2,3: RETURN
2530 DRAW"BM30,105;U50H20D90": P
AINT(27,100),2,3: RETURN
2540 LINE(DS,50)-(DS+30,95),PSET
,B: PAINT(DS+2,52),2,3: RETURN
2550 GOSUB 2650: CIRCLE(200,80),

```

```

12,4: PAINT(200,80),4,4: LINE(1,
120)-(1,120),PSET: FOR V=1 TO 25
5 STEP RND(15)+10: N=N+1
2560 N=N+1: IF N/2=INT(N/2) THEN
V1=80+RND(20) ELSE V1=100+ RND(
20)
2570 LINE-(V,V1),PSET: NEXT: LIN
E-(254,120),PSET: PAINT(2,134),3
,3: DRAW"C2;BM40,100;U30R10D10R3
0U10R10D30L50": PAINT(43,98),2,2

2580 A$="IT IS DAWN AS YOU ESCAP
E LURKLEY MANOR. YOU ARE VERY HA
PPY": GOSUB 1530: GOTO 2580
2590 GOSUB 2600: GOSUB 2650: GOS
UB 1600: A$="IT'S IGOR! HARUMPH!
NEVER HOPED-- I MEAN EXPECTED T
O SEE YOU AGAIN.": GOSUB 1530: A
$="I GIVE YOU A PARACHUTE. USE I
T WELL. I NOW RETURN YOU TO THE
GREAT HALL.": GOSUB 1530: GOTO 1
60
2600 A$="UP THE CHIMNEY YOU FIND
THE ATTIC. SOME WEIRDNESS HAPPE
NS - A COLOR FIELD...": GOSUB 15
30: FOR V=1 TO 20: PCLS RND(3)+1
: SCREEN RND(2),1: NEXT: GOSUB 2
650: RETURN
2610 PLAY "T1": GOSUB 1820: GOTO
2610
2620 LINE(S1,P1)-(S2,P2),PRESET,
BF: RETURN
2630 IF KZ=1 THEN KZ=0: RETURN E
LSE FOR T=1 TO 500: NEXT: COLOR
1,1: LINE(0,140)-(255,Y+12),PRES
ET,BF
2640 X=1: Y=140: COLOR 3,1: RETU
RN
2650 PMODE 3,1: COLOR 3,0: PCLS1
: SCREEN 1,1: LINE(0,0)-(255,135
),PSET,B: RETURN
2660 X$=INKEY$: HY=INSTR(DIG,X$)
: IF X$="" OR HY=0 THEN 2660 ELS
E RETURN
2670 FOR V=1 TO 15: CIRCLE(22,60
),V,RND(2)+2,1.8: NEXT V: RETURN

2680 REM - END OF LISTING

```

"WE ARE NEW"
WE ARE FOR YOU

SOFTWARE
AND
HARDWARE
FOR YOU AND YOUR
COLOR COMPUTER

CATALOGUE AVAILABLE



YOUR COCO DOWN
AND OUT LATELY ?
"YOU NEED"

DRACONIAN	\$30:50
SR-71	\$30:50
BUZZ BAIT	\$30:50
MR. DIG	\$30:50
CRASH	\$28:00
KING TUT	\$30:50
GRAND PRI	\$34:50
WILLY'S W.	\$34:00



Get A 'Boot' Out Of OS-9

By Charles Robitaille

Have you recently purchased OS-9 from Radio Shack and found that its system master disk wasn't directly bootable? Do you wish it was? Then read on! The reason you can't directly boot to your system disk is because you have Version 1.0 Disk BASIC ROM, instead of the newer Version 1.1. The newer version has the boot routine built into it and if you had this ROM you would only have to insert the system master disk, type DOS, and be on your way.

There are, fortunately, several ways around this. You could, for one, upgrade to the newer ROM, which isn't exactly cheap. You could wait for Radio Shack to send you a version upgrade patch and hope it includes direct booting. Or, you could do it yourself by following the simple procedure outlined in this article. Should you decide to do it yourself, you'll find you will have used only three sectors of disk space, and your system will seem much more professional.

My only caution is that if you have modified your original system master disk, then this procedure might not work for you. I would further add that this original system master disk should never be modified, except under instructions from Radio Shack, or you could lose compatibility with other users.

We will be working with a backup of the OS-9 system master disk, so if you haven't already done so, write-protect your original system master disk, Version 01.00.00. This is to protect it from being written to inadvertently. If you're not familiar with how to make a backup, I suggest you study Chapter 4 of the *Getting Started with OS-9* manual.

Once the backup is made, put away your original system master disk until it is specifically asked for. When your backup is made, leave it in drive 0 and delete two files from it. The files are named *LIST* and *OS9GEN*. They are occupying the three sectors we will need for our boot program. Don't worry, though, because the last thing we will do is copy these files back from the original. At that time, they'll occupy different sectors, but will operate just fine. By the way, to delete these files all you need to type at the OS9: prompt is:

```
DEL /D0/CMDS/LIST  
DEL /D0/CMDS/OS9GEN
```

After this is done, we will need to write a couple of short BASIC programs, one of them being the *BOOT* program itself. So, power down your system and power it back up again to get under control of Disk BASIC. Your backup system master disk should be in drive 0. Type in Listing 1 and then *RUN* it.

This program will set up Track 17, Sector 2 as Disk BASIC's granule allocation table. By doing this, we have told Disk BASIC that the only granule left to write to begins at Track 16, Sector 1, which is where we want our *BOOT* program to reside. It will then set up Track 17, Sector 3 to accept a directory entry.

The last thing it does is set up Track 0, Sector 2, which is OS-9's cluster allocation table. We will have, in effect, told OS-9 that it cannot use the above mentioned sectors because they are already in use. This program not only makes it possible to run the *BOOT* program, but it also protects the *BOOT* program from an inadvertent *SAVE* when under Disk BASIC, while also protecting it from OS-9 writing over it.

The next step in this procedure is the most pleasurable, because when we are done, we will have a bootable OS-9 system master disk. Type in Listing 2, which is the *BOOT* program. With the backup disk still in drive 0, type the following:

SAVE"BOOT"

You now have a bootable OS-9 system master disk. But, don't throw away your OS-9 Boot disk because it contains your drive speed test program which you will want to run from time to time. We must now get back the two files we deleted earlier. To get back into OS-9 simply type:

RUN"BOOT"

Insert your original write-protected system master disk and copy these two files to the bootable backup disk we just made. At the OS9: prompt, you'll need to type:

```
COPY /D0/CMDS/LIST /D0/  
CMDS/LIST -S #16K
```

```
COPY /D0/CMDS/OS9GEN /D0/  
CMDS/OS9GEN -S #16K
```

That's all there is to it. If you choose, you can make a backup of this disk, although it's not really necessary since if you somehow "bombed" it out, you could easily make another. By the way, the drive speed test program is named *CHR\$(130)+CHR\$(129)+".BIN"*. So, if you wanted to rename this program to *SPDTST.BIN*, for example, you would type under Disk BASIC:

```
RENAME CHR$(130)+CHR$(129)+  
".BIN" TO "SPDTST.BIN"
```

The program will be a lot easier to call up and send to another disk should you choose to use your OS-9 Boot disk

for something else.

OS-9 is one of the most powerful operating systems in the world for a microcomputer. It may very well be the

most powerful of all, so to all of you who have become part of the OS-9 network, I say congratulations. You have done your homework well and

spent your money wisely. Learn OS-9 and master it, and if this little boot program will help you enjoy OS-9 more, then your time will have been well spent.

Listing 1:

```
5 DEFUSR0=&HD66C:POKE&HEA,3:POKE
&HEB,0:POKE&HEC,17:POKE&HED,2:PO
KE&HEE,&H50:POKE&HEF,0
10 FORI=&H5000 TO &H5100:POKEI,&
HC9:NEXTI
15 POKE&H5020,&HFF
20 A=USR(0)
25 FORI=&H5000 TO &H5100:POKEI,&
HFF:NEXTI
30 POKE&HED,3
35 A=USR(0)
40 POKE&HEA,2:POKE&HEC,0:POKE&HE
D,2
```

```
45 A=USR(0)
50 POKE&H5024,&HFF:POKE&H5026,&H
D9:POKE&HEA,3
55 A=USR(0)
60 NEW
```

Listing 2:

```
5 DEFUSR0=&HD66C:S=1:H=&H27
10 POKE&HEA,2:POKE&HEB,0:POKE&HE
C,34:POKE&HED,S:POKE&HEE,H:POKE&
HEF,0
15 A=USR(0)
20 S=S+1:H=H+1
25 IFS<19 THEN 10
30 EXEC&H270C
```

OS-9 UTILITY Tidy Up Listings With LISTFILE

By Gerry Schechter

If you do any programming at all, you already know it can be very helpful to have a hard copy listing available in order to assist in debugging and modifying programs. When using Extended BASIC, a simple *LLIST* will provide the desired listing. In BASIC09, a *LIST procname >/p* will also do the job.

In either case, however, there are no top and bottom page margins, so you wind up printing on the perforations of the paper, which is kind of sloppy. If you ever break up your listings, punch holes in them and file them away as I do, then you know this can be a problem. Even if you don't, wouldn't it be nice to have a neatly formatted listing?

Since I like to have things simplified, I came up with a program called *LISTFILE*. This program, which is written in BASIC09, will produce a neatly formatted listing of any OS-9 file that

would normally be *LISTable*. Although it is designed with BASIC09 programs in mind, it can also be used with most other OS-9 files as well, with no problem.

Aside from skipping over the perforations, it will also print out today's date, the filename in elongated printer mode, and a page number at the top of every page for better readability and documentation. For BASIC09 programs, the listing is indented five spaces on each line that does not have a line number, in a manner similar to that of BASIC09's *LIST* command.

Using this program could not be easier. When it prompts you for the filename to be listed, all you have to do is to enter its name. This name must follow the OS-9 pathname conventions, so if the file to be listed is in your current data directory, all you need to do is to enter the filename. If it is not, you must enter the full pathname.

When your file has been listed, you will again be prompted for a filename to be listed. If you have no more files to be listed, just hit ENTER, and the program will end by itself.

The only part of this program you might have to modify has to do with putting your printer into the elongated print mode. As written, it is designed for the Radio Shack Line Printer VIII. Between lines 120 and 130 (not every line in numbered in BASIC09, you will find the necessary comments to help you in modifying the program to suit your particular printer.

I hope you will find this program useful, and that it will encourage you to write your own utility programs when the need arises. For those of you who don't like typing in programs, I would be happy to send you a copy of my program for \$1, if you will send a blank disk and a SASE to: 75 Midland Terrace, Yonkers, NY 10704.

The listing:

```
PROCEDURE LISTFILE
(* ----- *)
(* LIST FILE UTILITY *)
(* GERRY SCHECHTER *)
(* 75 MIDLAND TERRACE *)
(* YONKERS, NY 10704 *)
(* V1.1 - APRIL 1984 *)
(* ----- *)
```

```
PRINT CHR$(12)
PRINT " ** LIST FILE UTILITY **"
ON ERROR GOTO 130
DIM XX:INTEGER
DIM KEY:STRING[1]
DIM FILENAME:STRING[24]
DIM FIRSTTIME,BASIC:BOOLEAN
DIM INDATA,TEMP:STRING[128]
```

```
DIM LINECOUNT,PAGECOUNT:BYTE
DIM DISK,PRINTER,POINTER:BYTE
100 FIRSTTIME=TRUE
PRINT
PRINT " ENTER FILE NAME TO BE LISTED:"
INPUT " ",FILENAME
IF LEN(FILENAME)=0 THEN
PRINT
```

Martha Says....

Graham chose to report on the recent Australian Personal Computer Show in CoCo this month. As I went too, he asked if I'd summarize my reactions to the show in Rainbow.

Well firstly I think it stinks that Graham and others were there specifically to show the Education Dept people a CoCo or two, and there wasn't one to be seen anywhere.

This Tandy lot seem to think that they can pretend that CoCo is a real computer when it suits them, and then they call it a toy when they want to show off to the big boys!

Every other contender for the education contracts in NSW was there - Commodore, Micro Bee, .. well them anyway, but we weren't represented.

Tandy discovered a long time ago that people love the CoCo. They even admit candidly that they don't mind taking the money from the sales of the CoCo. Tandy still haven't discovered that it is OK to admit to having one in the stock line, when they are outside playing with IBM.

The show was hard to get into, especially if one had the tickets that were handed around as inducements by the various merchants. But once in it was immediately apparent that this was no ordinary exhibition! There were four floors of it and at times the number of people, and the smoke in the air, made it quite breathless.

There was a weak attempt at providing some food for sale, and I felt that the aisles could have been wider - especially where Control Data had this ridiculous 'robot'

running around, bumping into things and generally making a nuisance of itself. In fact it was a remote control toy fitted with a two way radio. And the operator was rude.

There were more MS DOS machines there than I ever hope to see again. I find it amazing that everyone has to run out and copy what is really for this day and age, a very average computer. Many of these MS DOS computers are headed for the scrapheap fast. They are like the multiple brands of TVs that disappeared soon after colour TV first reared it's ugly head in Australia.

The Tandy 1000 was 'launched' (their word) at the show, and will succeed. It is good, in that it does what it claims to do, that is, it emulates an IBM PC or PC jnr, but more importantly, it will be supported by a company with enough outlets to service the thing properly.

Graham has said that we will be supporting it, but where we get the people to do the work, I don't know. Knowing Graham, he'll do it - somehow, so I suppose that has to be another big plus in the 1000's favour.

All in all, I would have been happier back at my desk, the crowds of big cities are not the place for me.

Ron Foley wrote to Graham about THAT letter in March CoCo. I told Graham not to print it. Ron was absolutely right, it was filthy and therefore not fit for publication.

```
PRINT " PROCEDURE CONCLUDED."
PRINT
END
ENDIF
OPEN #DISK,FILENAME:READ
PRINT
INPUT " LINE UP PAPER AND HIT ENTER..",KEY
PRINT
OPEN #PRINTER,"/P":WRITE
110 READ #DISK,INDATA
IF FIRSTTIME THEN
GOSUB 120
FIRSTTIME=FALSE
IF LEFT$(INDATA,9)="PROCEDURE" THEN
BASIC=TRUE
ELSE
BASIC=FALSE
ENDIF
ENDIF
IF BASIC THEN
IF LEFT$(INDATA,9)="PROCEDURE" THEN
INDATA="*** "+INDATA
ELSE
IF LEFT$(INDATA,1)>"9" OR LEFT$(INDATA,1)="(" THEN
INDATA=" "+INDATA
ELSE
POINTER=SUBSTR(" ",INDATA)
TEMP=LEFT$(INDATA,POINTER)+
INDATA=LEFT$(TEMP,5)+RIGHT$(INDATA,LEN(INDATA)-POINTER)
ENDIF
ENDIF
ENDIF
IF LINECOUNT>60 THEN
GOSUB 120
ENDIF
PRINT #PRINTER,INDATA
IF EOF(#DISK) THEN
April, 1985
```

```
CLOSE #DISK
CLOSE #PRINTER
GOTO 100
ENDIF
LINECOUNT=LINECOUNT+1
GOTO 110
120 IF NOT(FIRSTTIME) THEN
FOR XX=1 TO 5
PRINT #PRINTER
NEXT XX
ELSE
PAGECOUNT=1
ENDIF
PRINT #PRINTER,LEFT$(DATE$,8);
(* START ELONGATION FOR LP VIII *)
PRINT #PRINTER,CHR$(27); CHR$(14);
PRINT #PRINTER,TAB((50-LEN(FILENAME))/2); FILENAME;
(* END ELONGATION FOR LP VIII *)
PRINT #PRINTER,CHR$(27); CHR$(15);
PRINT #PRINTER,TAB(50); "PAGE NO: "; PAGECOUNT
PRINT #PRINTER
LINECOUNT=2
PAGECOUNT=PAGECOUNT+1
RETURN
130 ERRNO=ERR
PRINT
IF ERRNO=215 OR ERRNO=216 THEN
PRINT " FILE NOT FOUND!"
GOTO 100
ELSE
PRINT " ERROR # "; ERRNO; " ENCOUNTERED!"
ENDIF
PRINT
PRINT " PROCEDURE TERMINATED"
PRINT " DUE TO ABOVE ERROR."
PRINT
END
```

From P 5

4. MS DOS (1000)

Paul Humphries.

5. Interfacing using CoCoConnection, Basic and Forth
John Poxon.

Final details are yet to be completed on BASIC for Beginners, and the two tutorials on Education. (We expect to have a pre-eminent speaker on this subject.) And finally, Lonnie Falk from American Rainbow will also be here to discuss "Aspects in the Development of the Color Computer". If this sounds a bit nebulous, it's because he wants to touch on "a bit of history, a bit of what is present, and a look into the future"!

Lonnie's address will probably be made over dinner on Saturday night, so there is no fear of missing him because you want to be at another tutorial.

Registrations.

PLEASE if you are coming, let us know. Frankly, although a lot have said on the phone that they are coming, I'm getting worried because there are only a few bookings.

In any case, the closing date for bookings will be 28th May. Why? Because we need time to order materials and in particular, the caterers need warning to prepare the food.

I think it will be an exciting time for us all. I am keen to see some of these high scorers from Australian CoCo's

'Scoreboard' go through their paces on the the Sunday. That is when we'll be throwing CoCoConf open so that you can see the latest gear, and talk to the software suppliers. It is also when the games competition will be underway!

So come on, people are coming from all over the place, Lonnie is coming from Kentucky, Tino is coming from Hobart, we know of folk who are coming from Gove in the NT, Melbourne, Adelaide, Perth, and Sydney - you won't be the only one making the long journey - register NOW! WE NEED TO KNOW YOU ARE COMING!

Transport.

We have had every kind offer of the use of a bus during the period of CoCoConf. It would be available to bring folk from Brisbane to the coast, and to transport folk around the coast to and from the various places that they will be staying. There is a cost involved, but it is only the cost of the fuel, so it's likely to be about \$4 to get from Brisbane to the coast, be ferried around all weekend, and returned to Brisbane. The cost for being taken from your lodgings to CoCoConf and back will be much lower.

If you would like to avail yourself of this offer, please phone Annette on 075-51-0015, and let her know.

COCOCONF

15 - 16 JUNE, 1985

- * TUTORIALS
- * FREE ticket to the Computer Expo
- * See and operate the latest in Hard and Software.
- * Pick up a bargain.
- * Catch up on old friends.

PROGRAM

SAT:

- 9.00 AM Welcome!
- 9.30 AM Tutorials. Choice of 4, or head off to the Computer Expo.
- 11.00 AM Morning Tea.
- 11.30 AM Return to Tutorials.
- 12.30 PM Lunch.

SUN:

- 10.00 AM Spend today with the Software Agents. Try out the new Programs, or join in the games contests.

9.00 AM Rotary Hall
Lawson St
Southport, Qld.

- 2.00 PM Tutorials. Choice of 4, or head off to the computer Expo.
- 3.30 PM Afternoon Tea.
- 4.00 PM Return to Tutorials.
- 5.00 PM Break to prepare for Dinner.
- 8.00 PM Dinner (Venue to be announced).

COCOCONF

Name:.....

Address:.....

.....

..... PC.....

Please bill my Mastercard / Visa / Bankcard NO

Please find Cash / M.O. / Cheque enclosed. Signed.....



I enclose full price \$ 39.95

I enclose part price \$ 9.95

and will pay the rest off before CoCoConf.

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 \$3.95 including postage and packing. Please enclose your cheque/postal order made payable to: Graham Morphet, PO Box 1742, Southport, 4215.

Subscription Rates

AUSTRALIAN
CoCo/MiCo/softgold

- \$3.45
 \$19
 \$31

Latest per copy
6 months
12 months

AUSTRALIAN
RAINBOW

- \$3.95
 \$21.50
 \$35

- \$3 per copy
1st issue '83

Feb '85 onwards \$3.95

Aug '84-Jan '85 \$3.25

to July '84 \$3.00

Back Issues

MiCo —
First Issue Oct. '83

CoCo/MiCo
First Issue Aug. '84

BOOKS

Byte \$5.95

Elementary

Help SORRY

OUT OF STOCK

Medium

Facts \$11.95

Advanced

MiCoHelp \$9.95

Medium

MiCo Exposed

V. Advanced \$11.50

DISKS & TAPES

Rainbow on Tape (program listings) \$12 for month of
Please note that RAINBOW ON TAPE is issued irregularly

Tape Monthly

CoCoOz

Latest \$8

MiCoOz

Blank tapes 12 for \$18 or \$1.70 ea

6 months \$42

Cassette Cases 10 for \$5

12 months \$75

Disks — \$3.50 ea. 10 for \$28.99

BULLETIN BOARD

CoCo Link —
Annual Sub. \$29

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

VISA BANKCARD

Cardholder

- CASH
 CHEQUE
 POSTAL ORDER



New Subscription Renewal

CUT OUT AND MAIL OR PHONE BANKCARD NO. (075 51 0015)

CUT OUT AND MAIL OR PHONE BANKCARD NO. (075 51 0015)

RAINBOW WARE

This range of clothing is made here on the Gold Coast

DESCRIPTION	SIZES	PRICE
V-Neck Crew Neck	12 14 S. M. L. XL	14.95
Tank Tops	10 12 14 S. M. L. XI	13.85
Tennis Dress	S. M. L.	26.20
Hooded Zip-up Jacket	S. M. L.	31.95
Ladies Hooded Pullover Top	S. M. L.	28.50
Ladies Shorts	S. M. L.	17.95
Children's Tank Tops	6. 8. 10. 12. 14	12.80
Children's T-Shirts	8. 10. 12. 14	14.20

COLOURS AVAILABLE white, pink, aqua, sky blue, yellow, hot pink, black

PLEASE SEND

ORDER FORM

_____ \$ _____
 _____ \$ _____
 _____ \$ _____
 _____ \$ _____

NAME _____

Complete the section below with one letter.
 Digits or space per square

INITIALS _____
 SURNAME _____
 ADDRESS _____

Subscription No

- CASH
 CHEQUE
 POSTAL ORDER

VISA BANKCARD

Cardholder _____

I UNDERSTAND THAT I CAN RETURN THESE GOODS WITHIN 7 DAYS FOR A FULL REFUND FOR ANY REASON

Of Back Issues, Tapes and Things.

With the exception of Nov 82, we have copies of all back issues available, and in fact need to reduce our stocks of many of them. The early copies of Rainbow are a source of excellent information for the new CoCo owner. The later copies reflect the growing knowledge of the average user of the time. There are games, utilities, hints, and programs for educational, business and club use. There are also many tutorials and articles of interest.

We also have considerable numbers of GoCo Magazine. If you don't have a full set of GoCo's, give me a call!

For those who want to complete their Rainbow collections, we are offering a one for three deal. Buy any three pre August 1984 Rainbows during February, and we'll give you one more of your choice free!

CoCoOz and MiCoOz this Month.

Australian CoCo magazine is choc full this month and this is reflected in the record number of programs on CoCoOz. For a start, the OSB section is represented by two programs, then there are several games, including Roo Hunt, and then we have a program which helps you find the amateur radio satellites. As this was written, the final number was unknown, but looks like there could easily be 15 programs for you to learn from, this month on CoCoOz #25.

MiCoOz maintains the quality showing of the last few months. There is a quiz by Tom Lehane, a couple of excellent games by Crazy John Day, Graham Pollock presents his first machine language program (a little bottler!), Michael Turk is as clever as ever with a listing formatter, and 13yr old Matthew Wiltshire takes us on a trip in space. Many of these programs run unchanged on 16K standard CoCo's - the trick is that you have to type them in - but then that never hurt anyone!

ANNOUNCING The BEST of CoCoOz!!

To assist teachers and others who are involved with children in learning situations, we have compiled a 14 program tape (or disk) which reflects some of the better Educational programming.

Programs include Quizzes on Flags and Rivers, the classic "Fractut", a fractions tutor, and "Taxman", a program which teaches Factors. Many of our best Writers are represented and we fully recommend this tape to Educators with CoCos who can't decide what to do with them!

'The Best of CoCoOz' is available for \$10.00 on tape, or \$21.95 on disk, postage paid.

The CoCoConnection

Connect your CoCo to the outside world.

Control Robots, Models, Alarms, Lighting Systems, Solar Panels for water or electrical generation, or create your own special use.

There are two models.

Mark I is available now and has eight reporting lines, and eight outputs.

Mark II to be released later in the year, has 4 inputs, and 12 outputs, and can accept the ADDA pack. ADDA packs provide 4 additional inputs, and 12 additional outputs. CoCoConnection will accept a maximum of 16 ADDA packs.

CoCoConnection comes complete with a driver program which you activate from your own programs.

PRICES:

MARK I	\$180.00
MARK II	T.B.A.
ADDA PACK	T.B.A.

AVAILABLE FROM:

AUSTRALIAN RAINBOW
BLAXLAND COMPUTER CENTRE
CASULA HOBBIES

Please allow 3 weeks for delivery.

Casula Hobbies

AUSTRALIAN MODEL RAILWAY SPECIALISTS

We stock:

- * Australian Rainbow and Australian CoCo magazines.
- * CoCoConnection - connect your train layout, burgular alarm, robot or model cars to CoCo.
- * Model Railways of Australian, American and European Prototype.

CASULA HOBBIES
P.O. Box 72,
LIVERPOOL. 2179.

SHOP
245 NORTHUMBERLAND ST.,
LIVERPOOL. 2170

I HAVE SOMETHING TO SAY!

COCO - CASSETTE



Every month you will receive a cassette tape with 10 ready to run documented programs.

1 YR (12 ISSUES) .	\$ 75 - 00
6 MO (6 ISSUES) .	\$ 39 - 00
SINGLE COPIES .	\$ 7 - 50

..Many programs use machine language!..

NEW * SOFTWARE * NEW

For the TRS80 Colour Computer
this month's *Feature Program*

KINGDOM OF BASHAN
A 32K adventure from
OWLS NEST SOFTWARE

Postpaid \$ 17.95
available at

Independent Tandy Electronic Dealers

OR GIVE US A CALL OR WRITE TO

COMPUTER HUT SOFTWARE

21 Williams St., Bowen, Qld., 4805 - Phone (077) 862220

ToTo Advertising



Box 5730,

Gold Coast Mail Centre, Qld,
4217 or phone (075) 39 2003



COMPUTERWARE FOR MICROS.

P.O. Box 81
Brooklyn Park, N.A. 5032
Phone (08) 336 6588

Peter and Jillian Collison, 11 Grantley Ave, Rostrevor, Sth Aust. 5073. (08) 3366588.

LET'S GET DOWN TO BUSINESS

PRO-COLOR FILE *Enhanced* 2.0 \$79.95

An all new version of PRO-COLOR-FILE will once again leave its mark as the most flexible database in its price range for the Color Computer.

• 60 Data Fields • 1020 BYTE RECORDS • TRUE MULTI DRIVE SUPPORT
• 4000+ RECORD CAPACITY • 4 USER DEFINED DATA ENTRY SCREENS • 28 MATH EQUATIONS • IF-THEN-ELSE FUNCTIONS IN EQUATIONS • FILE-WIDE RECALCULATION • 8 USER DEFINED REPORT FORMATS • 6 USER DEFINED LABEL FORMATS • TOTAL FIELDS ON REPORTS • SUMMARIZE FIELDS • SEND REPORTS TO PRINTER, SCREEN OR TEXT FILE • FAST ML SORT (750 RECORDS IN LESS THAN 5 MINUTES) • CREATE UP TO 16 INDEXES FOR SORTING OR REPORTING RECORDS • AUTO KEY REPEAT • KEYBOARD CLICK • STORES FORMATS FOR REPEATED USE • CUSTOM SELECTION MENUS • PASSWORD PROTECTION • CREATES FILES COMPATIBLE WITH DYNACALC®

Because of PRO-COLOR-FILES ability to send reports to a text file, this means you can use your favorite communications program to transmit reports to other computers or read them in with your favorite word processor for creating customized reports. You can also convert ASCII files from your favorite spread sheet program into data files that can be accessed for further reporting and analyzing. PRO-COLOR-FILE is also supported by the PRO-COLOR-FILE National Users Group with quarterly newsletters. Join the rest of the world and discover for yourself what you've been missing.

MASTER DESIGN \$39.95

This graphics program does more for you than just hi-res graphic editing. It will generate lettering in hi-res graphics that can be different sizes, skinny, bold, textured, drop shadowed, raise shadowed or tall. It will also interface with the Telewriter-64 word processor for printing hi-res displays with your letters.

As a graphics editor, it takes full advantage of all the extended BASIC hi-res graphic commands. Create boxes, circles, lines, copy displays and utilize GET and PUT features. Some added commands include mirror reflection, turn displays backwards or upside down, Squish displays, create dot patterns for shading or diagonal lines for creative backgrounds.

Special text files created with the Letter Head Utility allow you to access hi-res graphics from Telewriter-64, your own BASIC programs or PRO-COLOR-FORMS.

MASTER DESIGN comes with its own screen dump routine which interfaces with all popular dot matrix printers that have dot addressable graphic ability.

See reviews in:
July '84 Rainbow, Oct. '84 Hot CoCo Telewriter-64 © 1983 by Cognitec

PRO-COLOR-BUSI \$199.95

As most Businessmen do not want to learn how to run a Database and format it, we have Established a small business package. This package has, RETAIL INVOICE, WHOLESAL INVOICE, ACCOUNTS PAYABLE, REPORTS, LABEL, Plus a supplementary set of instructions to walk a newcomer through. The package includes the full PRO-COLOR-SERIES as well, so that, when the need arises, you can have someone build your own special Database. We will be establishing further up and running Databases as the need arises.

DYNACALC® \$129.95

This is the CoCo version of VISICALC. DYNACALC is not a toy. Sort up to eleven times faster than other top line spreadsheets. If you are really serious about using your CoCo for Business then there is no other spreadsheet to consider.

DYNACALC is designed to run with PRO-COLOUR-SERIES.

TELEWRITER-64 DISK — \$75.00 CASSETTE — \$66.00

COMPUTERWARE researched the word processors available for the Color Computer. This is the best. Telewriter-64 is a truly sophisticated system that is marvelously easy to use. It works with any 16K, 32K or 64K system and any CoCo compatible printer.

TOP-RATED COCO WORD PROCESSOR

PRO-COLOR-FORMS 2.0 \$39.95

PRO-COLOR-FORMS will access data files you create with PRO-COLOR-FILE and merge them with a letter or place them on pre-printed forms such as statements. Any field of information from your data file can be placed anywhere and repeated as many times in the letter or on the form. You can use the built in ML text editor for creating the form or use your favorite word processor.

• DESIGN UP TO 6 FORMATS AT ONE TIME • USER DEFINED PAGE SIZE
• SUPPORTS SPECIAL PRINTER CONTROL CODES • RIGHT JUSTIFICATION • PASSWORD PROTECTION • MERGES WITH GRAPHICS FROM MASTER DESIGN

If you use our graphics program MASTER DESIGN, you can merge graphics with your forms for added enhancements. Have your graphic letter head printed at the top of each letter or incorporate designs, bar graphs or any display created within the form itself.

PRO-COLOR-DIR \$24.95

Need to organize all your diskettes so you know where each program is? PRO-COLOR-DIR will read your directories and create a master data file that can be accessed by PRO-COLOR-FILE for sorting and reporting. 1000+ records can be stored on one diskette with valuable information about each program.

• DISK ID NAME • FILENAME/EXT • TYPE OF FILE • DATE CREATED • DATE UPDATED • NUMBER OF GRANS ALLOCATED • NUMBER OF SECTORS ALLOCATED AND USED • MACHINE LANGUAGE ADDRESSES

A diskettes directory can be re-stored in the data file with old entries deleted and new ones appended automatically. You can obtain hard copies of the information and create labels of the filenames for placing on the diskette itself.



COMPUTERWARE FOR MICROS.



P.O. Box 81
Brooklyn Park, S.A 5032
Phone (08) 336 6588

BACK IT UP!

\$59.95

(Supplied on Disk)

SUPER BACK-UP UTILITY®

SUPER BACK-UP UTILITY WILL PERFORM ALL OF THE FOLLOWING FUNCTIONS

1. TAPE TO TAPE (Regardless of most protection schemes)
2. TAPE TO DISK (Move Cassette programs to Disk)
3. AUTO RELOCATE (For those Cassette programs that conflict WITH Disk operating systems)
4. DISK TO TAPE (Place Disk programs onto Cassette)
5. DISK TO DISK (Our powerful Spit-N-Image Program. Regardless of protection schemes)

- MENU DRIVEN!
- REQUIRES 32K EXTENDED COCO
- REQUIRES 1 OR 2 DRIVES (For Disk Functions)
- ALL MACHINE LANGUAGE!!!

COMPARE WITH OTHER INDIVIDUAL PROGRAMS COSTING IN EXCESS OF \$100.00 OR MORE!!!

COLORPACK

\$39.95

ROM/RAM pack. (specify configuration)

COLOR QUAVER

\$29.95

Software Music Synthesizer on tape (requires 32/64K).

COLOR BURNER

\$99.95

EPROM Programmer (2716/32/32A/64/64A/128, 68764/66) with software.

LOWERKIT III

\$89.95

- Full-time upper and lowercase installs in 15 minutes.
- Normal and reverse video standard.
- Fully compatible with all Alpha and Graphic modes.
- Assembled and tested.

Important! Specify Color Computer or Color Computer II.

DOUBLE DOS II

\$44.95

(DISK ONLY) 64K required

Double DOS II—Now use 35, 40, or 80 track (double or single sided) drives, all on one system, all at the same time. All regular disk commands are supported with Double DOS II and are totally transparent to your BASIC programs! You can get up to 158 granules on a disk using an 80 track drive. These are the added commands:

- BAUD 1-6 ... change the BAUD rate.
- TRACK 35, 36, 40, 80 ... change number of tracks.
- DOUBLE ... enable the double sided option.
- PDIR ... print your directory to printer.
- DUMP ON/OFF ... send programs without a terminal program.
- RATE 6,35 ... change the head stepping rate.
- VIDEO ON/OFF ... reverse video without a hardware mod.
- SCROLL 1-255 ... change your screen scrolling speed.
- COMMAND ... will list all new commands.
- DUP 0, 1, 2 ... will allow copy & backup from one side of a drive to another!
- DATE ... you can enter the month, day and year as an extension to your programs when they are displayed during a DIR command.

We guarantee that this program will work using the above commands, with all types of 35, 40 or 80 track drives!

DRIVE O PACKAGES

\$549

\$7 shpg

More storage, Less cost!

... Our single-sided disk package gives 23,040 bytes more for a dollar less!

179, 712 Bytes: DD-1 SSD Drive
DC-1 40 Track Controller
CA-1 Cable

GRAPHICOM PART II Introductory pack **\$39.95**

Graphicom Part II is a video processing package that provides many functions that are missing in Graphicom. Here are just a few of the features provided by Graphicom Part II:

ENLARGE/REDUCE/ROTATE

Enlarge or reduce any portion of a screen by any amount, just like a photographic enlarger! Independent of the enlargement or reduction, rotate by any degree or fraction of a degree about any point on the screen.

PAN & ZOOM

"Zoom in" x2, x4, or x8 on any portion of the screen to do fine pixel work. Allows editing of Graphicom character sets with ease!

TYPESETTER & FONT EDITOR

Add text in 16 different sizes with several display modes to choose from including COLORED FOREGROUND & BACKGROUND text! Edit 8x8 characters for use in the typesetter. Over 30 character sets supplied on disk. "GRAB" function allows transfer of some Graphicom character sets to Graphicom Part II format.

PIXEL BLASTER

Allows the user to easily substitute or remove colors. Widen lines, swap BLUE & RED without effecting BLACK & WHITE, etc.

GRAPHICOM PART II DOES NOT REQUIRE GRAPHICOM TO RUN!

Graphicom Part II requires a 64K extended disk basic system, it will load and save both standard BIN files and Graphicom screens, and supports 1 to 4 disk drives with keyboard or joystick (analog or switch type). All functions support color or Hi-Res operation, as well as 4 screen display modes.

AUTOTERM

URNS YOUR COLOR COMPUTER INTO THE

WORLD'S SMARTEST TERMINAL!

YOU'LL ALSO USE AUTOTERM FOR SIMPLE WORD PROCESSING & RECORD KEEPING

EASY TO USE

ON-SCREEN EDITING via cursor. Full prompting and error checking. Key Beep and Error Beebop. Scroll bkwd/fwd white on line. Save/load files while on line. Maintain a disk copy of session. Automatic graphics. True lower case. Screen widths of 32, 40, 42, 50, 64. No split words on screen/printer. Print all or part of text. Search for strings. Well written manual goes step-by-step and has many KSM examples. Back cover is a cheat sheet.

RECOMMEND 32K to 64K

PLEASANTLY POWERFUL

Total communications ability, 128 ASCII chars. 1200 baud, etc. Send text, graphics, BASIS, ML Scan/Edit current data while receiving more data. Any modem. Fully supports D.C. Hayes and others. any printer, page size, margins, etc. Override narrow text width of received data. Examine/change parameters, KSMs and disk directories at any time. Handles files which are larger than memory.

TRULY AUTOMATIC

Create, edit, print, save and load Keystroke Multipliers (KSMs). KSMs automate almost any activity. dial via modem, sign-on, interact, sign-off. Perform entire session. Act as a message taker. KSM may include parameter changes, disk operations, editing, time delays, looping, execution of other KSMs, waiting for part-specified responses, branching based upon responses.

CASSETTE—\$69.95

DISKETTE—\$69.95

Nrth Old Colour Software, 9 Durham Court, KIRWAN, TOWNSVILLE, QLD. Phone (077) 73-2064.

Rainbow Valley Computers, RMB 6680, MAFFRA, VIC. Phone (051) 454-315

Blaxland Computer Service, P.O. BOX 2774, BLAXLAND, N.S.W. Phone (047) 39-3903

Paris Radio Electronics, 161 Bunnerong Road, KINGSFORD, N.S.W. Phone (02) 344-9111.

Geoff Tolputt, P.O. Box 140, WOOLONGABBA, QLD. Phone (07) 446084.

Crystal Blade Software, P.O. Box 256, ROSEVILLE, N.S.W. Phone (02) 467-1619.

The Computer Hut, 21 William Street, BOWEN, QLD. Phone (077) 86-2220.

COMPUTERWARE FOR MICROS

Peter and Jillian Collison

11 Grantley Avenue,

Rostrevor S.A. 5073

PHONE: (08) 336 6588

user group CONTACTS

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

ADELAIDE	JOHN HAINES 08 278 3560	GOLD COAST	SHERYL BENTICK 075-39-2803	PORT PIRIE	KEVIN GOMAN 084 32 1368
ALBURY	KEN DUNCAN 060 43 1031	GOSFORD	PETER SEIFERT 043 32 7874	RINGBOND	ANGELA RAVALLINGS 03 726 6521
ARNDHALLE	COLIN LEHMAN 051 57 1545	GULLBURN VALLEY	TONY HILLIS 058 59 2251	ROCKHAMPTON	KEITH SINNEN 079 28 6162
BALLARAT	MARK BELFANDER 053 32 6733	GREYKNICHES	BETTY LITTLE 08 261 4083	ROSEVILLE	KEN UZZELL 02 467 1619
BLACKHATER	KEITH BALLAGHER 02-627-4627	HERVEY BAY	LESLEY HODGKIN 071 22 4989	SALE	BRYAN McHUGH 051 44 4792
BLACKLAND	BRUCE SULLIVAN 047 39 3903	HOBART	HILTON ROBE 07 281 4059	SEACONCH HTS	MARK KIRBELL 07 269 5090
BOEN	TONY EVANS 077 86 2220	IPSWICH	PAUL MALONEY 069 24 1860	SMYTHESDALE	TONY PATTERSON 053 42 8815
BRAESSALL	BOB UNSWORTH 07 201 8659	JUNEE	GRAHAM BUTCHER 07 376 3400	SPRINGBOND	DAVID SEMMONS 047 51 2107
BRIGHTON	GLEN DAVIES 08 296 7477	KENMORE	CHRIS WAGLE 069 53 2969	STURT	MART DAVIS 08 296 7477
BRISBANE EAST	ROB THOMPSON 07 848 5512	LEETON	STUART PATYNER 043 51 4214	SUNBURY	JACK SMIT 03 744 1355
BRISBANE SA	GRAHAM BUTCHER 07 376 3400	LITHGOM	LENNIE DUGGAN 02-607-3791	SUTHERLAND	IAN ANNEBEL 02 528 3391
BRISBANE WEST	BRIAN DOUGAN 07 30 2872	LIVERPOOL	LEN MALONEY 079511333782	SYDNEY EAST	BOB JONES 02-331-4621
BUNDABERG	JIM McPHERSON 071 72 8329	MACKAY	ROBIN ZIUELLIS 03 450211x445	SYDNEY TEENS	ROD HOSKINSON 02 48 5948
CAMPBELLTOWN	LEO GINLEY 02 605 4572	MALDEN	MAX HUCKERBY 051 45 4315	TAMBOURTH	ROBERT WEBB 067 65 7256
CAMBERRA	SHAWN WILSON 042 51 2339	MARYBOROUGH	LYN DAWSON 049 49 8144	TAMBOURTH	GARY SILVESTER 046 81 9318
CAULFIELD	JEFF SHERN 03 528 3724	MELBOURNE	JEFF SHERN 03 528 3724	TONGALLA	TONY HILLIS 058 59 2251
CHARTSMOOD	BILL O'DONNELL 02 411 3336	MELBOURNE	SCOTT HEADLON 050 23 6016	TONGALLA	ARTHUR SLADE 02 622 8940
CHURCHILL	GEORGE SPIDART 051 22 1389	MILDURA	STEPHEN SEIPLE 051 27 6841	TONGALLA	NICK CARTER 076 35 6911
COLLINGTON TEENS	DAVINE MANNION 02 623 5805	MORFETTSVALE	KEN RICHARDS 08 384 4503	TONGALLA	DAVID PROUT 076 32 7533
COOMA	ROSS PATT 0448 23 045	MORFETTSVALE	ALE BATE 067 52 2465	TONGALLA	ADVANCED
DAVENDONG	DAVID HERRICKS 03 793 5157	MORFETTSVALE	GEORGE FRANKS 051 34 5175	TONGALLA	BRUCE KING 069 25 3091
DARWIN	BRENTON PRIOR 089 81 3014	MORFETTSVALE	BRIAN STONE 063-72-1958	TONGALLA	ATHELE SHART 02 848 8830
DENILIQUIN	WAYNE PATTERSON 058 81 3014	MORFETTSVALE	PETER ANGEL 071 68 1628	TONGALLA	CHRIS HINTER 086 45 3395
DUBBO	GRAEGE CLARKE 048 89 2095	MORFETTSVALE	LYN DAWSON 049 49 8144	TONGALLA	BRIAN McGOWLEY 042 71 4285
DUNEDIN	LEIGH EVANS 059 68 3392	MORFETTSVALE	ROBBIE DALZELL 08 386 1647	TONGALLA	PAT KERNODE 056 74 4583
FORSTER	GARY BALLEE 045 54 5029	MORFETTSVALE	ROY LOPEZ 044 48 7031	TONGALLA	SPECIAL INTEREST GROUPS
FRANKFON	BOB HAYTER 03 783 9748	MORFETTSVALE	DAVID SNALL 068 62 2682	TONGALLA	BRISBANE 059
GIPPSLAND STH	PAT KERNODE 056 74 4583	MORFETTSVALE	TON LEWKE 047-31-5303	TONGALLA	JACK FRICKER 07 262 8869
GLASTONE ALBERT	VAN GORKUM 079 72 2353	MORFETTSVALE	IAN MACLEOD 09 448 2136	TONGALLA	STUART HALL 08 361 1922
		MORFETTSVALE	MARTIN WELLS 02 449 2077	TONGALLA	BLAXLAND 059
		MORFETTSVALE	RON LALOR 045 83 8223	TONGALLA	BLAXLAND 128K
		MORFETTSVALE	ROB DALZELL 08 386 1647	TONGALLA	808 THOMPSON 047 30 2468
		MORFETTSVALE		TONGALLA	TIM SHAWK 079 28 1846
		MORFETTSVALE		TONGALLA	ROCKHAMPTON MCo
		MORFETTSVALE		TONGALLA	STONEY MCo
		MORFETTSVALE		TONGALLA	PAULA VILAY 02

THE RAINBOW
Registered by Australia Post-
Registration No. QBG 4009
Graham Morphett,
P.O.Box 1742, Southport, QLD 4215.

AUSTRALIAN
The only other magazine you need for
your TRS 80 Color Computer.



Which ever way you look at it...

POSTAGE
PAID
AUSTRALIA