For your TANDY Color Computer

Registered by Australian Post — Publication No. QBG 4009

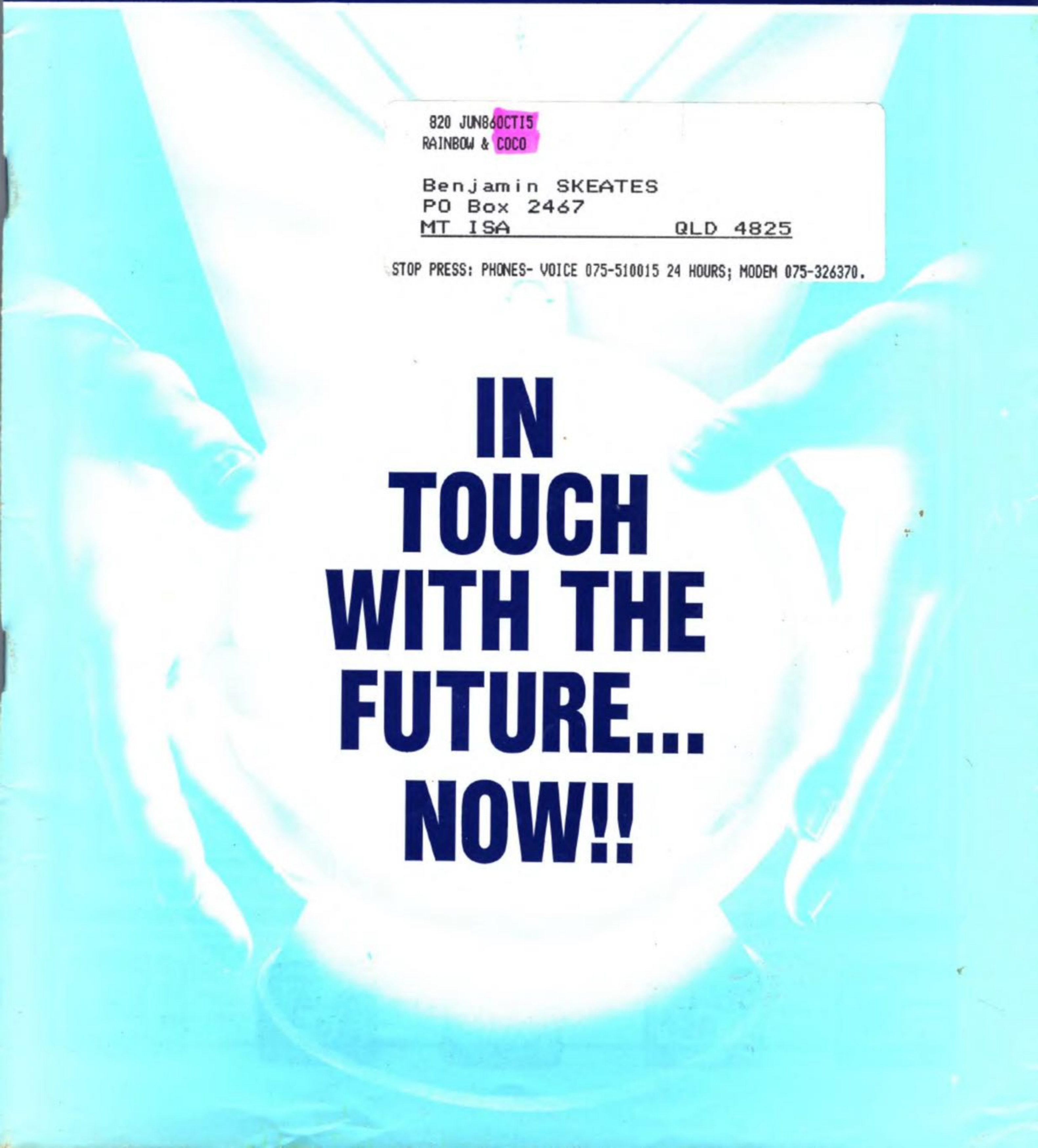
NEW GUINEA K5.95

$4.50

AUSTRALIAN RAINBOW

October, 1985

No.52



IN TOUCH WITH THE FUTURE... NOW!!

COMPUTERWARE FOR MICROS.

bankcard welcome here

VISA

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

B-DOS ©

USER FRIENDLY DISK OPERATING SYSTEM NOW AVAILABLE ON DISK FOR YOUR CO-CO

AUTO (line number) :	35-40 TRACKS
ERROR TRAPPING :	BAUD (value)
COLD (cold start) :	PDIR (print dir)
PCLEAR (16 pages) :	SWAP (var1, var2)
OS9/DOS (included) :	UNNEW............

A COMPREHENSIVE MANUAL + MUCH MORE

* * * ONLY $44.95 * * *

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 • CUS' SELECTION MENUS • PASSWORD PROTECTION • CREATES ES COMPATIBLE WITH DYNACALC® •

SUPER BACK UP UTILITY ©

. . . WITH S.B.U. FROM COMPUTIZE YOU'LL NEVER NEED ANOTHER BACK-UP UTILITY FOR YOUR CO-CO!!!

1. **Tape to Tape**
2. **Tape to Disk**
3. **Auto Relocate**
4. **Disk to Tape**
5. **Disk to Disk**

- **Menu Driven!**
- **Requires 32K Extended Co-Co**
- **Requires 1 or 2 Drives**
- **All Machine Language!!!**

• • • ONLY $59.95 • • •
(SUPPLIED ON DISK)

My Co-Co Does What!

• LOWER CASE KIT •

TRUE LOWER CASE PLUS REVERSE VIDEO

Now with dual 5:7 and 7:9 characters. Use your COLOR BURNER to put in your own special character sets. (optional)

For visual comfort and Pro programming send for LOWERKIT II-C . . . **$89.95**

• COLOR BURNER •

An EPROM programmer is the perfect tool for creating your own program packs. The COLOR BURNER programs the most popular erasable, programmable eproms: 2716(2K), 2732(4K), 2764(8K), 27128(16K) and 68764/66(8K).

EASY TO USE - STEP BY STEP INSTRUCTIONS.

•• ONLY $99.95 ••

** COLOR QUAVER **

THE ULTIMATE ALL-SOFTWARE MUSIC SYNTHESIZER!

COLOR QUAVER, an amazing Music experience from your Color Computer.

FEATURES:

- Real electronic music synthesis that is more than bleeps.
- Full four part harmony, all in precise tempered tuning.
- Five usable octaves, variable tempos, rhythms from 32nd note to whole note.
- Fast compiler provides finished music in five seconds . . . up to 250 notes per voice line, 1000 notes in all.
- Over 40 pages of instructions, explanations, hints, listings and samples.

•• INCREDIBLE VALUE AT ONLY . . . $39.95 •• (tape only)

** CoCo MAX **

THE HIGHLY ACCLAIMED CoCo MAX
•• NOW AVAILABLE ON CASSETTE ••

Simply the most incredible graphic and text creation "system" you've ever seen. You will be generating Hi Res images in minutes.

You don't explain CoCo Max, it does that for itself!

So if you think you can't draw, use CoCo MAX and be truly amazed at the results.

SYSTEM REQUIRED: 64K COCO
DISK OR CASSETTE
A STANDARD JOYSTICK
MOUSE OR KOALA PAD.

TRUE VALUE AT $149.95 . . .

ADD $4.00 HANDLING CHARGE WITH EACH ORDER

PARIS RADIO ELECTRONICS

CALL
(02) 344 9111 VOICE
(02) 344 9511 BBS
TO ORDER

CoCo OS-9™FLEX ™

Free Diskette with Each $50 Purchase

COMPILERS

K-BASIC — A BASIC language to MACHINE language Compiler; includes an Extended Macro Assembler. CCF — $292.00

PL/9 — by Graham Trott. A combination Editor/Compiler/Debugger; Structured Programming at the "almost Assembly Language" Level.
CCF — $292.00

INTROL C Compiler — Full Featured C Compiler, Linking Loader includes full Library Manager CCF — $549.00

•Forth — Forth language for CoCo CCR — $99.95
CCF — $99.95

MICROWARE BASIC09 — for CoCo OS-9 systems. CCO — $149.95

MICROWARE C Compiler — for CoCo OS-9 Systems. CCO — $149.95

DYNA C — C Compiler from the authors of Dynastar and Dynaforms.
CCO — $99.95
CCF — $99.95

DATA BASE'S

XDMS — "Mainframe" User's say: "We don't have anything NEARLY as powerful as XDMS;" pure Assembly Language, F-A-S-T and small enough to operate on a single-sided 5" disk.
XDMS (V) I — CCF — $199.95 **XDMS (V) II** — CCF — $299.95
XDMS (V) III — CCF — $$399.95

RMS — Machine Language Data Base Manager — Super Fast.
CCF — $220.00
CCO — $180.00

OPERATING SYSTEMS

XEX — The latest Flex Operating System from Frank Hogg Laboratory USA. Includes hi res screen drivers, user definable keyboard, supports 128k upgrades, and 35, 40, and 80 track drives. CCF — $149.95

OS-9 for the CoCo. The hotest selling software to hit Australia.
CCO — $99.95

DISASSEMBLERS

SUPER SLEUTH — Interactive; extremely POWERFUL!! Disk File Binary/ASCII Examine/Change, Absolute or FULL Disassembly, XREF Generator, Label "Name Changer," and Files of "Standard Label Names" for different Operating Systems included. CCF, Obj. Only **$89.95**
CCF, w/Source **$145.00** CCO, Obj. Only **$89.95**

DYNAMITE + — Excellent standard "Batch Mode" Disassembler Includes XREF Generator and "Standard Label Names" Files.
CCF, Obj. Only **$89.95** CCO, Obj. Only **$89.95**

WORD PROCESSING

STYLOGRAPH III — A full Screen-oriented WORD PROCESSOR (what you see is what you get); also supports Daisy Wheel proportional printers and WORDPAK CCF — $165.00
CCO — $165.00

STYLO-SPELL Spelling checker to suit stylo. CCF — $99.95
CCO — $99.95

STYLO-MAIL Mail merge program for Stylograph. CCF — $89.95
CCO — $89.95

DYNASTAR & DYNAFORMS — Another full screen Editor and Word Processor for the CoCo. CCF — $149.95
CCO — $149.95

XWORD — A powerful word processor that can be run under O-PAK, XSCREEN hi res drivers and WORD-PAK 80. CCO — $99.95

SPREAD SHEETS

DYNACALC — CoCo's best and fastest Spread Sheet system.
CCR — $149.95

UTILITIES

UTILIX is a unix like utilities package for OS-9. It includes 15 different utilities to aid you in the manipulation of text files. CCO — $75.95

FILTERS KIT No. 1 — Eleven utilities used as filters for OS-9
CCO — $39.95

FILTERS KIT No. 2 — Ten more utilities, APPEND, CONFIRM, FF FORCERROR, MACGEN, NULDEVICE, REP, SIZE, TOUCH, and unload.
CCO — $39.95

SDISK and BOOTFIX — Use 40 or 80 track, single or double sided drives with OS-9. Create a bootable double sided system disk. CCO — $59.95

HACKERS KIT No. 1 — This package contains a set of programs useful to anyone working with assembly language, trying to unravel the operating system or other assembly language programmes, or customizing their own.
CCO — $59.95

XSCREEN — Hi Res Screen driver for OS-9 CCO — $39.95

O-PAK — Hi Res Screen driver for OS-9 CCO — $56.95

SEARCH AND RESCUE — disk utilities to search and retrieve lost data.
CCO — $56.95

XTERM — A communications program for use with XSCREEN.
CCO — $82.95

MEMORY MINDER — Disk diagnostics program. The most comprehensive program available to analyse your disk drives
DOUBLE SIDED CCR — $149.95
SINGLE SIDED CCR — $139.95

XMENU — This is a program that creates a menu driven environment for the coco under OS-9. CCO — $69.95

RAM-DISK — This program simulates a disk drive, utilizing spare RAM memory. Very fast execution of commands. For use with 64K or 128K memory, Flex, XEX, or OS-9. CCF — $49.95
CCO — $49.95

MAGAZINES and BOOKS

AMERICAN RAINBOW — Direct from the USA the same month of publication for the same price as the newsagents. Why wait! **$6.80**

AMERICAN HOT COCO — Airfreighted direct each month. Subscription available. **$5.80**

THE OFFICIAL BASIC09 TOUR GUIDE **$29.95**

HARDWARE

PBJ WORKPAK 80 — an 80 column card to fit in the expansion port of the CoCo. Will run under Flex or OS-9. **$199.95**

PBJ WORDPAK II — The ultimate wordpak. Includes smooth scrolling, software controlled video switch, improved character set, hardware inverse video and up to 8K video RAM. **$249.95**

PBJ CC-BUS — A six slot software selectable expansion bus. Will take your disk controller, WORDPAK 80, GAMES Cards etc. **$219.95**

PBJ PC-PAK — This is a dual function cartridge which contains a centronics compatible parallel printer port and/or a battery backed real time clock.
w/ real time clock $199.95
w/o real time clock $ 99.95

PBJ 2SP-PAK — A 2 port RS-232 serial interface with programmable baud rates up to 19.2K. Each port can also be interrupt driven. **$119.95**

DISK CONTROLLERS

J&M Disk Controller with JDOS Disk Basic	$219.95
RADIO SHACK Disk Controller Ver 1.1	$199.95

MISC

CC-VID Monochrome video output board	$ 35.00
MC6809EP Processor chip	$ 24.00
MC6883 SAM chip	$ 24.00
4164-150 nSec, 64K chips	each $ 7.80

VISA

TM FLEX is a trademark of Technical Consultants
TM OS9 is a trademark of Microware
PRICES SUBJECT TO ALTERATION WITHOUT NOTICE

PARIS RADIO ELECTRONICS
161 BUNNERONG RD, KINGSFORD 2032
P.O. BOX 380, DARLINGHURST 2010
Ph: 344 9111

Availability Legends —
CCF —' Color Computer FLEX
CCO — Color Computer OS-9
CCR — Color Computer RS-DOS

PRICES DO NOT INCLUDE POSTAGE AND PACKAGING

AUSTRALIAN
RAINBOW

index

UPPER CASE = ARTICLE + PROGRAM
lower case = article only

AUSTRALIAN RAINBOW Publisher and Editor Graham Morphett. Co- editor Kevin Mischewski. Assistant Editor Sonya Young. With grateful assistance from Brian Dougan, Richard and Judy, Bob Thomson, Paul Humphries, Alex Hartmann, Michael Horn, Jim and Sheryl Bentick, Annette Morphett. Cover Art Jim Bentick.
ADVERTISING DEADLINES: The 7th of the preceeding month of publication. All advertising is arranged through ToToAdvertising, PO Box 6730, Gold Coast Mail Centre, Qld. 4217
OS 9. Kevin Holmes is the contact for os 9 information. He also has access to OS 9 Software from the US. His address is: 39 Pearson St., Narara, NSW. 2250.
All programmes in this issue of AUSTRALIAN RAINBOW are available on Rainbow on Tape.

Telephone: 075 51 0015 Voice; 075 32 6370 CoCoLink
Printed by: Australian Rainbow Magazine PO Box 1742. Southport, Qld. 4215
Founder GREG WILSON.

PRINT #-2,

If we seem to be forever saying "thanks" in this magazine, it's because the magazine is very much a combined effort which really depends on a lot of people.

This was demonstrated last month when one by one, everyone even remotely connected with the preparation of the magazine fell victim to the flu.

The magazine schedules fell back and back and we were panicking! But despite severe illnesses, Jim, Sheryl, Annette, Sonya and Kev all did their bit and we made it!

Speaking of people, you'll notice a new voice on the phone. Janet Stott has joined us and will be largely responsible for the processing of your orders and your letters.

As many have told us - this area of our magazine's work is not at all fast - that's going to change!

I have had an opportunity to take a closer look at EARS - the Speech Recognition unit.

It is startling to realise the computer is recognising what one says - very much shades of Hollywood - but it's here today and for real.

You can program by voice, instruct the computer to operate an appliance, or just tell it to "FIRE" when the baddies come on the screen!

I have other plans too - I am planning to explore more fully its use with visually and handicapped people and will report in due course.

stralian programs these days in
prove - let me know what you think.

If you haven't already done so, please fill out your survey forms found in last month's magazine and return ASAP - this will help us make you a better magazine.

LETTERS

Dear Graham,

May I suggest that, as a strongly-user oriented organization, you give consideration to the establishment of some sort of code of ethics for firms that advertise with you, perhaps similar to that required of its members by the Australian Direct Marketing Association.

I am not, incidently, suggesting that any of your advertisers are ever likely to be deliberately dishonest, or to willfully disregard the rights of their customers; however it is a fact that the CoCo supply industry has been historically very much an amateur affair, dominated by the user groups and self-helpers. Under such circumstances, the sort of commercial professionism in consumer protection practiced by the direct marketing majors has never been necessary. Quite simply, then, I am concerned that suppliers may emerge who simply do not know what are the consumers rights, and may accordingly violate them out of sheer ignorance.

Certainly, to the extent of my personal experience with direct mail CoCo marketers in Australia, I can say that none has even the faintest idea of how to run a successful direct mail operation. And while marketing competence cannot be taken as a complete index of willingness to look after the consumer, nevertheless I have found that the two are often related.

If you think that my idea has merit, may I suggest that, as a first step, you approach the A.D.M.A. - I am sure that they would be prepared to help you draught a suitable code. Then you could ask your advertisers to subscribe to it, so that, on the one hand, they would know what was expected of them, and on the other, that your readers would have a better idea of their rights with mail ordering of computer gear, and would be able to see that you were looking out for their interests.

What do you think?

I have written what I freely admit is a rather esoteric program to sort and analyse data concerning the number of householders in a postcode area. I use this to plan my own direct mail programmes. The Program, in BASIC, includes sort and select routines, a save and verify routine and a few other features that might be interesting. However I am stumped for any use to which it might be adapted that would be more broadly interesting than its present function.

Lamaan Whyte
PORTLAND, VIC.

Lamaan,

Like lots of ideas which have merit, I find myself thinking of yours that it is great - but how do I get the time to even follow it up with the letter.

You struck the nail on the head when you said our supply industry is small and home based.

Never the less, there is protection for buyers.

Where there is an instance of concern brought to our notice, we do follow it up.

But, at the same time, this is not an invitation to wingers. If I find an individual wasting our time on nothing, we act as decisively as with those suppliers who have done the wrong thing.

Graham.

*

Dear Graham,

Firstly may I congratulate you for the excellent job you and your team are doing in carrying on Greg's wonderful work. As a voracious reader I find both Australian Rainbow and Australian CoCo packed full of interesting and informative material.

I have a 64K ECB CoCo at home but spend a lot of time working on a Prime mainframe (with 2MB of RAM and 3 x 168MB hard disc drives). Despite the power of the Prime system, CoCo never ceases to amaze me with its power.

I found the article by Roy Lopez (Aust CoCo February, 1985 page 42) very interesting, and now understand the "FortyK" program I have been using (Aust Rainbow February, 1984 page 37) a little better. Roy has been bitten by 0808 in several places, but he does list the actual addresses correctly. This is a trap we all fall for at some time or another and to me it does not detract from the value of his article.

Reg Lang in his article "Have You Heard From Mico Lately?" (Aust Mico February, 1985 page 48) has been similarly bitten by failing to include the "all zeros" combination. Once again this does not detract from an excellent article which I am sure "tells it as it is" for CoCo as well as Mico, and in fact any computer - only the times will change - "what happens" is basically (no relation to cobolally) the same.

The program "A Good Sort" by John E. Allen (Aust Mico February, 1985 page 53) appears to be a useful utility. On first reading the listing appears to have errors at lines 1700 and 2480, but on keying it in to CoCo exactly as listed it worked perfectly. I was surprised to find that "IF...GOTO" worked the same as "IF...THEN", but to be consistent lines 1700 and 1900 should use the same convention. Also line 2480 could as easily read "GOTO 1100" to re-start the loop, thus saving memory by doing away with line 2500. I would like to offer some constructive advice on the use of arrays. John has used the zero element of his two arrays, which is a common practice which saves memory space, but has not used the "ten" element of the array. The statement "DIM A$(9),B$(9)" will give the required number of elements for this program and not waste memory, and the program will still run. I prefer not to use the zero element of an array unless memory space is critical.

In his letter (Aust Mico February, 1985 page 47, column 3) Harold Turner enquired about a line which forgot to tell us is that this form of IF... THEN is perfectly valid and useable. The syntax is "If true/false expression THEN action (ELSE action)". Since any numeric variable can equate to TRUE (non-zero) or FALSE (zero) the line in Harold's program will run without error and will increment the variable S if T has a positive or negative value, but will do nothing if T=0.

The following short program will demonstrate:-

```
10 DIMA (5)
20 FOR X = 1 TO 5
30 READ A(X)
40 NEXT
50 FOR X = 1 TO 5
60 PRINT A(X),
70 IF A(X) THEN PRINT "TRUE" ELSE PRINT "FALSE"
80 NEXT
90 DATA -30, -1,0,1,152
```

Of course in the case of Harold's program there is no way of telling whether anything is missing without seeing the listing, but readers may like to use this convention when testing for zero as a memory saver (and key-stroke saver). Of course we can also use "IF BOT T THEN first action (ELSE second action)" which will reverse the procedure (If T=0 then the first action will be taken else the second action). The IF NOT convention is a handy way of testing keyboard input for non-numeric entry. I make all keyboard INPUT statements string statements, to avoid the user "bombing" the program by entering a letter instead of a number (TM error). This is particularly useful in programs requiring numeric input where zero is an invalid response but no upper or lower level can be set.

```
Eg 100 INPUT "ENTER VALUE TO BE AD
   DED";(A$)
   110 A=VAL(A$): IF NOT THEN SOUN
   D 50,15:
   PRINT "I CAN'T ADD LETTER AND I
   WON'T WASTE MY TIME ADDING ZERO
   ":GOTO 100
   120 V=V+A
```

I found a bug in your accounting program (Aust Rainbow February, 1985 page 11) which you might like to fix. I first came across this problem while doing a course at Prime Education Centre. They use a similar Postcode/Statecode/State routine to produce a State name from the postcode. The problem is that people in Cooma and other surrounding areas do not live in the ACT. There is no easy way of getting all ACT postcodes, but I believe only the numbers from 2600 to about 2610 are in the ACT. You also include the Northern Territory as part of South Australia. I'm not sure if it applies to all NT towns, but 5700 to 5799 seems about right. I can't guarantee there are no SA towns in that block either. The following line will fix the NT:

```
1135 IFLEFT$(P$,2)="57"THENST$="NT"
```

As for the ACT, a 98% fix would probably be:

```
1105 IFLEFT$(P$,3)="260"THENST$="ACT"
```

I trust my comments throughout are taken in the vein intended, as helpful advice to my fellow users. I know all your contributors have been a great help to me, and I feel it's time I started passing on what little I know.

Carry on the great work.

Keith Lewis
PADSTOW, N.S.W.

Keith,

We use a more detailed postcode routine when we send subscription copies, and we've been gradually getting the other programs with this routine into line!

Graham.

*

Dear Graham,

I can imagine that picking up the pieces since you took over the business has not been easy. But it looks now as though you are up and away. Keep up the good work.

Now a comment or two, as constructive as I can make them.

I've had a CoCo Basic for the last two and a half years and have stoutly resisted all urgings to upgrade to ECB or 32K or what have you. The reason is that although my ambition is one day to plan Beethoven's last Piano Sonata, I prefer to stick with my tin whistle until I have mastered the basics of music. In other words, I find that, with the little time I have at my disposal for computer work, I have more than enough on my hands mastering my little CoCo: bigger things will have to wait. Then too, I am not at all interested in gee-whiz graphics or games: I am interested in CoCo as a COMPUTER. My perseverance has paid off in that, having mastered BASIC, I am now well into Assembly, and have even managed to start to make sense of the book THE FACTS which I bought from Greg a long time ago. (How I do wish someone who can write plain English for Assembly beginners would undertake to rewrite FACTS! It really is a dog's breakfast prepared by some untidy Einstein from his own cryptic and disordered notes. Bless him/her whoever he/she was for doing so much heavy spade work. But oh for the gift of clear communication).

You will understand then that my face falls as I skim through each issue of RAINBOW and see "PMODE", "32K DISK" ... Ho I bless the contributors whose listings are headed "16K or "4K". How I bless too the occasional contributor who prefaces Assembly listings with clear, intelligible comments.

I fully appreciate that all this puts me amoung a small minority of subscribers. I appreciate too that this minority could improve their lot by submitting material to suit their interest. (That includes me; so now that I have a little printer, I may indeed try sending in some little pieces. Wish me luck).

A suggestion. Would it be possible to include a small glossary for the uninitiated? Something like the one found in the June '85 "Your Computer" every so many issues would be most helpful. But I'm thinking too of each RAINBOW issue: a very brief synopsis on your "Contents" page indicating the nature of the article, plus ditto of such mysterious regular columns as "OS9" (some sort of

disk system, but what?).
Vic Kennedy
HAWTHORN, VIC.

Vic,

You're a long term reader so I wont bore you with too much reiteration, but for the benefit of new readers, we can only print what we get!

We especially pleasure in receiving programs from new users. Yes we know there are others who can do things better than you, but they cant teach others at your level as well as you can!

We need 16K Basic programs. Usually we place them in Australian CoCo, when available. There they have double application because the users of the MC 10 computer can often adapt them.

A glossery did indeed occur in the August '85 Rainbow, thanks to the very talented (and extremely handsome - so the girls tell me) Mark Rothwell.

Graham.

*

Dear Graham,

I typed in Penguin Patrol from April, 85's Aust Rainbow and ran it. It said Penguin Patrol by Apollo on the screen and played a song. When play was supposed to start Looie and the Cone-heads didn't move. It paused for a moment and played a tune again. Then it went back to text and displayed the score and high score (which were both nothing) and asked to press any button to play again. I tried deleting line No. 7 to stop the high speed POKE from going but it didn't work. What I want to ask is why won't Looie and Coneheads move?

What is your verdict?
Scott Herbert
PALM COVE, QLD.

Scott,

You need a Users' Group.

Graham.

*

Dear Graham,

I am writing to let other readers know that you can get into a disc drive for around $450.00.

I bought two Chinon F-051D bare drive's from Data Parts in Shepparton, Victoria, for $149.00 each and a power supply and case for $78.00. All I had to do was to screw the drives to the power supply and then plug the cables in to the drives. The power supply was all set to go on the 240V side.

I got my disk controller and cable from Software Spectrum for $276.00 as I like the added advantage of being able to use Jdos as well as Tandy Dos.

Beside the commands that come with the Tandy Dos, Jdos automatically numbers lines in basic programming, loads and executes the OS-9 and Flex functions to trap errors and maintan control over line number and error condition, a most valuable feature.

With Jdos you can run 40 track single or double side drive's 35 or 35 Radio Shack compatable. With my controller I can run either Tandy or Jdos, but with the above features I am always in Jdos. The disc controller and drives have gold plated edge connectors.

The whole project only took about 5 minutes to assemble and all that was needed was a screw driver and no prior experience. Specifications for the drive:

No of heads 1
No of tracks 40
Current 12V 0.2 amps
Capacity (MFM) 250K Bytes
Track to track 6MS
Average latency 100 MS
Rotational speed 300 RPM
Transfer rate (MFM) 250K
standby power dissipation 2.84 watts.

With the Software Spectrum controller, you get a manual for each operating system. So far, I have been very pleased with my system compared to $1,000.00 + from Tandy.

All the best for now.
Tony Hillis
KOYUGA, VIC.

*

Dear Graham,

I am writing on behalf of my son Matthew, who has been subscribing to Rainbow Magazine for the past six months. Matthew is thirteen years old and has a Tandy TRS 80 64K computer. It was recommended that we subscribe to Rainbow, but I think perhaps it was a little advanced for him. Most of his difficulty seems to have come about when speed pokes are included in the programs - he has lost programs which he has been trying to save. Any help on this point would be appreciated.

Perhaps a subscription to Australian CoCo would be more appropriate. Do they include some shorter programmes in their issues?

I noticed in the last Rainbow that another edition of HELP is due out soon. Would this be of assistance to a beginner?

I look forward to your comments on the above.
Pamela Pittwood
WARRACKNABEAL, VIC.

Pamela,

I think it is an error to feed a new user Australian Rainbow, without Australian CoCo.

Rainbow has some simple stuff in it, but increasingly, it tends to address itself to those who have been using their computers for some time.

The new HELP was due out soon, but we had to shelve work for it this month due to our increased work load.

It should be out by Christmas, if William Flexgerald gets his Typesetter going by then!

It is nice to see a growing readership in Victoria - we hope to hear more from Warracknabeal!

Graham.

*

Dear Graham,

Hope all is well up your way.

Like many other Coco users who have the Telewriter-64 word processor, I have been continually impressed by its many capabilities. One of those capabilities is an option in the main menu which when pressed returns a word count.

Well, quite by accident I discovered that it is possible to get a partial word count. By typing (Clear) (E) (end block character) and placing the cursor directly after the block, returning to the menu and pressing (W) will return the number of words from the cursor to the end of the file.

Often essays will have word limits for each section of the whole assignment. This gives a convenient quick check in such a situation.

Also I am wondering if anyone has successfully combined the Telewriter-64 program and the function key program that comes with the upgrade HJL keyboard. If so, I would be very interested in hearing how you did it.

Tom Stuart
ARMIDALE, N.S.W.

*

Dear Graham,

How nice it was to go into my local newsagent and to see the Rainbow on the shelves.

With the passing of Greg Wilson I had lost touch and hope of getting a good mag such as the Rainbow again.

I have bought the July and August issues of Rainbow and I am impressed with what I have seen. I used to attend user meets in the past but my job situation has changed and I now work regular night shift which limits my time mostly at the key board and at any meetings. I am only poorly conversent in Basic and you are using other cumputer languages of which I am completely lost. My system comprises the 32K extended Basic computer, disk drive and DMP-200 printer.

I am interested in back issues where printer dump programs have been highlighted. Any advice on that topic will be received with great interest as I find printer programs hard to come by unless I know somone personally.

As a family we get quite a lot of pleasure from the TRS-80 - mostly my two boys who can beat me at any game I can devise which makes my games old hat to their modern minds. The only problem is they expect their Dad to write good games programs of which I don't have the time or know how. I would appreciate a mag tape which incorporates reasonably intelligent games for (smart bumbs) in the age group of 13 to 16 with the emphasis on smart going to the 16 year old, namely Anthony.

John Wren
NOBLE PARK, VIC.

John,

With any luck, this issue of Australian CoCo should have the program "G.S.P.R." It is a screen dump, first written by Paul Humphreys, and is one which has kicked around the Brisbane group for some years, gradually being added to as it went!

The Best of CoCoOz #2 part I (16K) and part II (32K ECB) are available. Part II has anumber of simulations and adventures which may be of some interest to Anthony.

Graham.

*

Dear Graham,

We own a TRS80 16K colour computer (not extended) and a DMP400 printer. We are looking for a program that will allow us to type numbers and letters in graphics. These are to be used to slot prices and product names into plastic edging fixed onto the front of shop shelving. (Some supermarkets now do this).

We also want to print handbills to advertise our shop and the shop name etc. should be in these enlarged letters. We had in mind that the lettering should be something like one inch or so.

If you have such a program avalable on cassette or even written out, would you please advise me the cost.

Janet Maessen
BAYLDON, N.S.W.

Janet,

The Graphicom series of programs could probably accomplish what you want to do.

You will need a disk drive however, so you might find it cheaper in the short term to buy the tape version of CoCoMax.

CoCoMax does not have the choice of fonts and the ability (at this stage) to enlarge them, but I understand that this may be possible in the future.

Graham.

*

Dear Graham,

Could you please advise me on methods currently available to expand CoCo's number of characters across the screen from the present 32. For example, what does an 80 column card do?

I would also like to congratulate you and your team for the great job you are doing in running Rainbow and Coco. I would expecially like to thank John Redmond for his marvellous articles on Forth.

John Vila
DEER PARK, VIC.

John,

There are a number of programs available, which allow the use of the graphics screen for text presentation.

We recently featured "Screen 51" in this magazine, and "Super Screen" and "Rainbow Writer" have been available from Software Spectrum for some time.

In addition, screens have been incorporated in a number of recent programs in this magazine.

For example, a screen which was developed by Bob Horne, I think again from a program by Paul Humphreys (don't shoot me if I'm wrong Bob!), is incorporated in the excellent series of educational programs by him which is currently being presented in Rainbow.

The 80 column card is another approach. It fits in the ROM port of your computer - usually on a multipack board or Y cable, and you connect your monitor to it.

The Word Pac II 80 column card is pretty fancy. It provides very smooth scrolling and can switch backwards and forwards between text screens and graphics screens - a very handy trick. The card allows excellent word processors such as Stylograph to be used in an 80 column format.

Graham

*

Dear Graham,

You wobbygong! Imagine tantalizing your readers with the exciting news that there could be a hardware mod to fix the lousy colour control in PMODEs 0, 2, and 4, and then holding back! (Then again, what do you expect from the author of OS8??).

In response to certain implications that I'm crazy I might just mention that I wasn't photographed blowing into Tino's ear!

I too, was at the Education expo early on Sunday morning. The Apple people may have been upset at you - they were livid at me!! For a start I secretly crashed a couple of demo programs on the IIe's, and replaced them with "visit Tandy" messages. I then listened to an Apple salesman trying to convince a housewife that she needed to by a xx-thousand dollar Macintosh for her kids benefit - cripes! She later saw me and asked for my opinion, so I did a little convincing about the CoCo. If looks could kill then the Apple salesman would've butchered me! The Tandy display was very BASIC (forgive the pun), then again, that they turned up at all was magnificient! I also went around all the stalls, collecting pamphlets, brochures, etc. to give me a basis for writing my own.

The RMC chipset described sets me alight with pleasure - the CoCo is nowhere near dead yet! I read the August review of the Commodore Amiga, and I personally think it is the best micro I have ever witnessed - including IBM PC, Macintosh, Tandy, anything! It uses the 68000 as well. Tandy are going to have to be good to beat this (perhaps the rumoured Model 9 could?)

Unfortunately you added lines 1-3 to my program in August Rainbow (the tribute to Greg), and this might cause problems for 16Kers.

I did not hog the computer at CoCoConf! There were at least six other CoCos all blazing away - besides which if I didn't move I got a good kick in the patoot from Andrew Simpson.

I agree with your statements about the "Conf". The very, very best bit was sitting across form Andrew, and watching his face when he was announced winner - on ya mate!

My dad was very secretive with construction of that project he sent in! He just showed it to me one day and said "try it". It worked beaut!

Enclosed are some drawings of the latest S8 devices:

- OS8 graphic tablet
- OS8 chips
- OS8 software
- OS8 koala pad
- OS8 SAM

Finally, I won't be able to touch the CoCo again until December - the marks from my last physics exam are reflective of the time I spend on the CoCo. School wins again! However, by January I should have an Australian graphics adventure that should entertain most people, plus some other niceties. Its a shame I had some great graphics ready for the competition. Could you extend the competition to January or is that too difficult? Don't worry about it if it is.

Little CoCo says to give a "good-on-ya" to Johanna Vagg, so here goes - good on ya Johanna!

Andrew White
Bracken Ridge. Qld.

P.S. Graham I have just read the article by Laurie O'Shea (Australian CoCo August, 1985), and I think it is brilliant. Urgently needed, and I want to do something of this manner. As I can't do computers until December, this is long term, but I think I can gain access to 16 CoCos at a local school. Would you also send, or publish detailed technical (hardware and software) information on E.A.R.S.

Andrew,

Yes, I too think my enduring memory of CoCoConf '85 will be the look on Andrew's face!

As much as I'd like to tell you more, I can't tell more about the colour thing, because the people concerned haven't told me! But they go to your Users' Group I'm told, so maybe you can find out!

Andrew, we want you to know that no one has implied that you are crazy. We thought it was common knowledge that you are!

Your drawings are most welcome and lend a little comic relief this month to a very advanced and usually dry subject. Many folk seem to think of OS8 as a bit of a joke - I don't really know why this should be, perhaps because it is a little complex?

The drawing competition in Australian CoCo will be extended till the February magazine. We have had trouble finding space for the feature, and some nice things are coming in - including a picture of a steam engine, which, as some readers will appreciate, is a winner before we look at the others!

If others in Brisbane want to start a care group, then we'll get them to contact you.

There will be more on EARS next month.

Johanna said to say hello to Little CoCo and to tell him he is welcome in Forbes any time - she needs a new door stop!

Graham.

*

Dear Graham,

I want specific info on how to set up a BBS. Hardware, Software, System, etc. I want to be able to auto answer, detect incoming Baud Rate and adjust the system to suit, be able to access Viatel, be able to send at either 300, 75, or 1200 baud. Not too impossible, maybe, but I'm not made of money!! (Who is). I guess the Avtek Modem is as close as I can get, but do you have any other suggestions.

Mini drives: ie. 3 or so inch disc systems. What value?

Super Coco. Is it a dream or have we got a good thing coming. I am seriously thinking of getting into a more capable system for my main work, but want to stay with a Unix capable system. The Japs have some good 6809 based gear according to the mod, but somehow I think I would like to stay within the Tandy System without getting into MS-DOS. Perhaps Atari will shake Tandy out of their lethargy.

You made mention of an affordable 256K upgrade coming. What news?

I would appreciate your short answers.

We had our first CUG meet here a couple of weeks ago. One of those 6 attending produced a Coco that really had sat under the bed for 18 months. Such a pleasure to get people underway. Just the simple things like loading, etc. The grin that tells you that "It's not so difficult afterall".

Ken Spong
YARRAWONGA, VIC.

Ken,

Don't ask me about bulletin boards! The one we've got needs drop kicking!

The difficulty is that software and hardware is usually designed in the US where it works beautifully. Bring it here and interface it with an Australian CoCo, and a Telecom phone line (as in one that doesn't have to compete with anyone else!) and you've got trouble!

Fortunately, Jacky at Paris Radio is working on a total package which should see some changes to this state of affairs. Some Bulletin Board operators have not had the problems we are experiencing, and they may have some clues, but for me - my advice is to keep it as simple as possible!

The Avtec modem does sound good though.

Mini drives are great if you have larger drives from which you can obtain programs. There is a small amount of software available from Software Spectrum on mini disk, but the price of the disks precludes most from using them at this stage.

I would like to think that they could be used in situations where children are involved, because they will cope with little fingers better. They can also be carried conveniently in a shirt pocket; this can be a good or bad thing.

Software Spectrum supplies a Mini drive for CoCo; you should talk to them if you need more info.

I am not at liberty to say much more about the so called Super CoCo.

It has to happen, but when and what its final configuration will be is unknown - even by Tandy!

The Atari system is very good, but is relatively expensive and unsupported. I believe you would be well advised to wait for Tandy to finalise their design, even if it takes a while CoCo can do virtually all the Atari does, with some minor exceptions (color, speed).

Paris Radio are the memory upgrade kings - talk to them about 256K - though they'll probably want to talk more about their upcoming 512K upgrade!

I'm glad you are enjoying your new meet group. There are two sides to being a meet contact.

There's always those who will make the job awful by being nasty. There are the ones who think they are going to come along and steal programs, and there are those who cause trouble.

But if you can cope with these problems, you can really do some good work with people - helping them learn about computing, and in the process, making their lives more worthwhile.

Graham.

*

Dear Graham,

I am thinking about getting a modem for my COCO II 16K ECB. Could you please write and tell me if the Avtec Mini Mode from Paris Radio Electronics will fit any telephone. And does it have the right baud rate to connect up to any system. And can the Compac Program from Tandy be used instead of a RS-232 Program Pack. And what will the Compac Program do.

I don't know much about modems at all so could you please write to me and tell me about the different ones on the market and will I need cables and things like that. Could you also tell me about hotlines that I could hook in to if I had a modem.

I wrote to you about a year ago about not being able to get an Assembler anywhere. Well Tandy are now restocked with EDTASM+ Cartidges and they can be ordered from any Tandy Store.

Could you please also tell me what the best terminal package is and where I can get it.

Martin Preston
COBDEN, VIC.

Martin,

The Avtec is a 300 Baud modem.

The phone numbers for various Bulletin Boards around Australia can be found in the visitor's section of CoCoLink, our own Bulletin Board, which you can call on 075-32-6370.

Terminal packages are often a matter of personal preference, and like many things, it is better if you can see several operate before you buy.

Fortunately in Melbourne, you have the Wilsons and the Millers to go to, to see these programs, not to mention some very good User Groups! Also ask the Tandy people to show you Compac.

Graham.

*

Dear Graham,

I have been having a bit of trouble with the "WEFAX" program that was featured in "Aust Rainbow, April, 1985". So far I have it decoding the Australian weather station ASM on 5100KHz but cannot get it to syncronize after centering the picture and returning the joystick to center bottom.

I have tried changing the value of F3 in line 50 but it seems to have no effect. I wonder if the problem is due to the program being written in America and the scan speed is set on their 60hz mains frequency. If anybody has this program running I would sure appreciate it if they could help me out.

While I'm at it I have up and running on the CoCo the programs "Pathcalc" and "Greyline" which were written for the Commodore 64 and featured in an Australian amateur radio magazine. Any other "Hams" interested can contact me QTH-R.

Ross Platt
C/o Tandy
CONDOBOLIN. N.S.W.

Ross,

"Wefax" caused some problems for some people, but not for others. The major one being that it is set up for an Epson printer.

Perhaps those who are experiencing no difficulties might like to let us in on their secret!

Graham.

OS8 MEMORY UPGRADE

by Barry Cawley

As promised, Barry has supplied a new module for OS8 this month which upgrades the memory of the CoCo.

The original plan was to provide a megabyte of memory, but while experimenting with OS8, Barry was able to add a second megabyte by implimenting the X/Mode 12 function, and flipping the Barque parameters.

In undertaking this work, he also was able to get improved error messaging.

Type the listing into the computer, SAVE it, then RUN. The BASIC routine is flipped into OS8 by the X/Mode 12 function and then you're away!

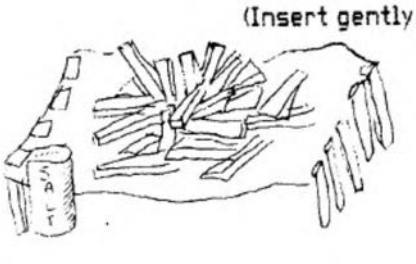
THE LISTING:

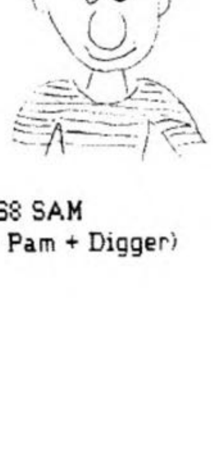
```
1 '******OS8 MEMORY UPGRADE*****
   ********BARRY CAWLEY*********
   *********10/9/85*************
2 GOTO10
3 SAVE"OS8:2":DIR2:STOP
10 POKE330,50:POKE331,98:POKE332
,28:POKE333,175:POKE113,0:POKE33
4,126:POKE335,173:POKE336,165:PO
KE410,126:POKE411,1:POKE412,74
11 DIMD$(15):Z=159:B$=""
12 CLS:FORI=1TO15:READD$(I):NEXT
13 IF PEEK(&HBC)<>6THEN 14 ELSEP
RINT"SUPER EXPANDED MEMORY MODUL
E    COPYRIGHT (C) 1985 OS-8
                    SOFTWARE"
14 PRINT"DISK SUPER EXPANDED MEM
ORY      MODULE COPYRIGHT (C) 19
99       OS-8 SOFTWARE"
15 GOTO 19
16 PRINTA$;:B$=B$+A$:IF A$<>CHR$
(13)THEN 15
17 PRINTA$:PRINT"OK"
18 GOTO15
19 PRINT CHR$(Z);
20 Z=Z+16:IF Z>255 THEN Z=159
21 FOR X=1TO15
22 A$=INKEY$:IF A$<>"" THENPRINT
CHR$(8);:GOTO25
23 NEXT
24 PRINT CHR$(8);:GOTO19
25 PRINT A$;
26 IF A$=CHR$(13) THEN 27 ELSE 2
9
27 IF B$="CLS" THEN CLS:GOTO19
28 IF B$="?MEM" OR B$="PRINT MEM
" OR B$="PRINTMEM" OR B$="PRINT
MEM" THEN B=MEM+200000:PRINTB:B$
="":GOTO19  ELSE 34
29 B$=B$+A$:GOTO 19
30 DATALIS,IS NO LONGER VALID PL
EASE READ  THE OS-8 MEMORY MODUL
E MANUAL
31 DATAPCL,IS AUTOMATICALLY SET
TO 200      GRAPHICS PAGES,RUN,IS
 ALTERED SEE THE               OS
-8 MEMORY MODULE MANUAL,EDI,IS N
OT A VALID COMMAND           PLEA
SE READ THE OS-8 MANULE,PEE,I'M
GETTING FED UP WITH ASKING  YOU
TO READ THE MANUAL
32 DATAONE OF US IS CONFUSSED I
THINK  MAYBE IT'S ME,SORRY I DO
NOT UNDERSTAND THAT
33 DATAARE YOU FEELING ALRIGHT.
THIS    INPUT IS NOT UP TO YOUR U
SUAL    STANDARD,TYPE THAT AGAIN
SAM,IF YOUR'RE FEELING ALRIGHT T
HEN I MUST BE SICK
34 Y=0:FOR Y=1 TO 9 STEP2:IF LEF
T$(B$,3)=D$(Y) THEN 36 ELSE NEXT
35 Y=RND(5):Y=Y+9
36 PRINTD$(Y+1);:B$="":C$="":X=0
:Y=0:PRINT:GOTO19
```

(See Andrew White's letter in the letters column for an explaination of these cartoons.)

OS8 TABLET.
(Not tonight dear ...)

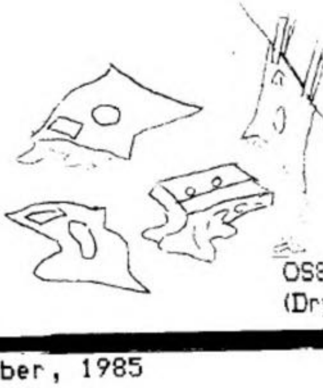
OS8 CHIP
(Insert gently.)





OS8 SAM
(+ Pam + Digger)

OS8 KOALA PAD.
(Hey Man ....)



OS8 Software.
(Dry clean only)

EDUCATION PAGE

After our big Education month last month, I thought we'd better let the others have a go this month so I'll keep this bit short.

There have been a number of teachers wanting to use the programs by Bob Horne with disk drive, so Bob supplied the following amendments which should be applied to 'Homophones' first so you get the drift of the adjustments necessary.

```
55 IF PEEK(188)=14 THEN A=14:PO
KE31920,14:POKE31965,197:POKE31
966,241 ELSE A=6
```

Also change the following lines:

140 POKE 246,A: Leave the rest of this line the same.

170 POKE 246,A+6: Leave the rest of this line the same.

380 POKE 246,A: Leave the rest of this line the same.

The program automatically adjusts itself for disk or cassette operation. MAKE SURE YOU SAVE THE ADJUSTED VERSION BEFORE RUNNING.

In 'Antonyms', add line 155:

```
155 IF PEEK(188)=14 THEN POKE 3
1920,14:POKE31965,197:POKE31966
,241
```

Once more our thanks goes to this man for his assistance - these programs of his are proving very popular, and this further advice is most welcome.

Bob's help this month has gelled a thought which has been growing for the past few months.

It would be very cute if we could just switch on a computer and have it do a job in the classroom, like they do with business computing.

But lesson preparation has never been like that, and I am doubtful that classroom computing will ever be like that either.

Right now, the only teachers which seem to be achieving anything with computers in schools are the ones who have involved themselves personally.

And after all, this is only logical - you don't expect to be prepared to teach a subject without some training in the area, so why expect to be able to use a computer in your classrooom, if you don't even know how to switch it on?

The ease of use of the computer has fooled some into thinking they don't need to learn - this is not so, and the sooner teachers recognise this, the better.

COMPOUND WORDS

by Bob Horne

The main graphics screen in "Compound Words" represents the inside of a partly underground bunker, complete with a viewing slot, monitors and control panel. The object of the game is to join two smaller words together to form a compound word. If you accomplish this at your first try, a rocket will take off on an inter-planetary exploration flight. If you get the word wrong, the rocket stays on the launch pad. However, you cannot go on to the next example until you get this one right.

Altogether, you get 10 questions. However, there is a timer which limits the game to between 6 and 8 minutes. With children using "hunt and peck" typing, I have found that most get through the 10 examples before the time runs out. If you wish to alter the timing here, then alter line 130 - where TIMER is divided by 100. Simply alter the "100" to a smaller number.

In any case, the timing section does not take effect until an answer has been typed in.

The next screen presents the correct words for the examples you missed. This is followed by a scorecard. If you had some wrong answers, the next screen presents the missed examples for you to do again. After this section is completed, it is time for a new game, or in the classroom, the next user.

The following is a program outline:

40: Pokes in the ML routine for a border on the text screen.

50 - 60: DRAW and GET TV monitor and rocket.

70: Sets the text screen to orange instead of green.

80 - 90: Present the title screen, then allow you the choice of wheather you need instructions or not.

120: Draws the letters on the screen.

130 - 140: Decreases the visible portion of the time line.

150 - 230: Gets up to 15 letters as a word input.

240 - 280: Separate the data elements.

290 - 340: Mix the elements in a random way.

380: Main loop begins.

400 - 410: Choose random example.

420 - 460: Draw the words on the screen.

510: The end of the main loop.

520 - 620: Correct answer routine

530: Takes the rocket to the top of the screen.

540: Makes the rocket gradually disappear.

630 - 640: Wrong answer routine.

650 - 740: Presents the words you had wrong. The data concerning these words is put into the arrays S3$(X) and S4$(x) for later use.

750 - 870: The scorecard routine.

880 - 1040: Practise the wrong ones.

1050 - 1100: Title screen - this is presented once only (using the flag N1 in line 10 and 100)

1110 - 1280: Instructions.

1290 - 1540: Draw the main screen.

1550 - 1560: Inkey routine.

1570 - 1650: Draw strings for the graphics screen.

1660 - 2160: Main data. Note the format:

WORD, CORRECT WORD TO JOIN ON, ANOTHER WORD, ANOTHER WORD.

If you change the data, then be careful here that you didn't present data where there could be 2 correct answers. Consider this line:

DATA EVERY, ONE / NEATH / BODY

The program will only accept the word "EVERYONE" as correct, but "EVERYBODY" would also be correct, but the program would say it was wrong.

2170: The data for the machine language "instant border". By the way, if you don't like my border (which is CHR$(249)) then change A=USR0(249) to some other character ie A=USR0(128) would put a solid black border around the screen.

Alterations would have to be made in lines 1080, 1130, 1180, 1220 and 1260.

I have found that for this game, the TV set needs adjusting severly so that the letters can be readable (partially to distinguish between the "N" and the "M").

THE LISTING:

```
1 REM**************************
2 REM*     COMPOUND WORDS      *
3 REM*      BY BOB HORNE       *
4 REM*      IPSWICH, QLD.      *
5 REM**************************
10 CLS0:CLEAR800,32730
20 PCLS:DIMCH$(90),S1$(51),S2$(5
1),S3$(10),S4$(10),A(51),R(10),R
D(10),TV(15),IN$(15)
30 FORX=1TO90:READCH$(X):NEXTX:F
ORX=1TO51:READS1$(X),S2$(X):NEXT
X
40 FORX=32731TO32759:READA:POKEX
,A:NEXTX:DEFUSR0=32731
50 PMODE4,1:COLOR1,0:PCLS:R$="UE
2F2DUH2UR2U7F2D3L2U6H2G2D6L2U3E2
D7R2":DRAW"S4BM2,14"+R$:GET(0,0)
-(8,14),R,G
60 COLOR0,1:PCLS:LINE(0,0)-(20,1
5),PSET,B:COLOR0:LINE(2,2)-(18,1
3),PSET,BF:DRAW"S3C1BM9,10"+R$+"
S4":LINE(2,10)-(18,13),PSET,BF:G
ET(0,0)-(20,15),TV,G
70 POKE359,57:SCREEN0,1:IF N1=1
THEN90
80 GOSUB1050
90 GOSUB1110
100 N1=1:Q=RND(-TIMER)
110 CLS0:PRINT@268,"stand"+CHR$(
128)+"by";:A=USR0(249):GOTO350
120 DRAW"BM"+STR$(X1)+","+STR$(Y
1):SL=LEN(W$):FORI=1TO SL:B$=MID
$(W$,I,1):C=ASC(B$)-32:DRAWCH$(C
)+"BR3"
125 IF B$="M" OR B$="m" OR B$="W
" OR B$="w" THEN X1=X1+4 ELSE X1
=X1+3
127 NEXTI:RETURN
130 IF FL=2 THENRETURN ELSE TI=I
NT(TIMER/100):TI=254-TI:IF TI<30
 THEN TI=30
140 COLOR1:LINE(TI,112)-(X3,117)
,PSET,BF:COLOR0:X3=TI:RETURN
150 FORX=1TO15:IN$(X)="":NEXTX:A
N$="":TIMER=T1:IN$=INKEY$
160 FORX=1TO15
170 IN$(X)=INKEY$:IF IN$(X)="" T
HENLINE(X1,Y1)-(X1+4,Y1-5),PSET,
BF:COLOR1:LINE(X1,Y1)-(X1+4,Y1-5
),PSET,BF:COLOR0:PLAY"T100L2002A
G":GOSUB130:GOTO170
180 IF IN$(X)=CHR$(8) THEN FORXX
=1TO15:IN$(XX)="":NEXTXX:COLOR1:
LINE(111,169)-(199,177),PSET,BF:
COLOR0:X1=113:GOTO150
190 IF IN$(X)=CHR$(13) THENRETUR
N
200 IF IN$(X)<"A" OR IN$(X)>"Z"
THEN170
210 W$=IN$(X):GOSUB120:X1=X1+3:A
N$=AN$+IN$(X)
220 NEXTX
230 RETURN
240 NA=1
250 LS=INSTR(AN$,"/"):IF LS=0 TH
EN A$(NA)=AN$:GOTO270
260 A$(NA)=LEFT$(AN$,LS-1):AN$=R
IGHT$(AN$,LEN(AN$)-LS):NA=NA+1:G
OTO250
270 SL=0
280 RETURN
290 FORC=3TO1STEP-1
300 Z5=RND(C)
310 IF Z5=1 AND SL=0 THEN SL=1:C
A=C
320 T$=A$(C):A$(C)=A$(Z5):A$(Z5)
=T$
330 NEXTC
340 RETURN
350 GOSUB1290:POKE359,126
360 FL=0:RA=0:TIMER=0:T1=0:R1=10
:R2=43:X2=90:X3=254:NQ=0
370 SCREEN1,1
380 FORZ=1TO10
390 NQ=NQ+1
400 Q=RND(51):IF A(Q)=1 OR A(Q)=
2 THEN400
410 A(Q)=1
420 AN$=S2$(Q):GOSUB240:CA$=S1$(
Q)+A$(1):GOSUB290
430 X1=108-6*LEN(S1$(Q)):Y1=143:
W$=S1$(Q):GOSUB120
440 FORX=1TO3:X1=163:Y1=113+X*15
:W$=A$(X):GOSUB120:NEXTX
450 X1=10:Y1=189:W$="TYPE THE CO
DE-WORD AND PRESS <<ENTER>>":GOS
UB120
460 X1=113:Y1=175:GOSUB150
470 T1=TIMER:COLOR1:LINE(1,183)-
(254,190),PSET,BF
480 IF AN$=CA$THEN GOSUB520 ELSE
 GOSUB630
490 IF FL=1 THEN450
500 IF TI=30 AND Z<10 THEN Z=10:
NEXTZ:PLAY"L1T1P4":GOTO650
510 NEXTZ:PLAY"L1T1P4":GOTO650
520 PLAY"T2V20L1603CEL804CL1603A
L404C"
530 IF FL=1 THEN560 ELSEPLAY"V31
T100L10001CGAFDEC":FORY=R2 TO1ST
EP-1:PUT(R1,Y)-(R1+8,Y+15),R,PSE
T:NEXTY
540 FORX=1TO15:GET(R1,2)-(R1+8,1
6),RD,G:PUT(R1,1)-(R1+8,15),RD,P
SET:NEXTX
550 RA=RA+1:COLOR0:X1=100:Y1=189
:W$="Rocket launched.":GOSUB120:
COLOR1:LINE(X2,100)-(X2+6,106),P
SET,BF:COLOR0
560 IF FL=1 THENCOLOR0:X1=5:Y1=1
89:W$="It's right - but rocket s
till misfires.":GOSUB120:FL=0
570 PLAY"T1L1P4":X2=X2+9:R1=R1+2
5
580 IF Z=10 THENPLAY"T1L1P4":RET
URN
590 COLOR1:LINE(111,169)-(199,17
7),PSET,BF:LINE(71,136)-(109,144
),PSET,BF
600 FORX=121TO151 STEP15:LINE(16
1,X)-(219,X+8),PSET,BF:NEXTX
610 LINE(1,183)-(254,191),PSET,B
F:COLOR0
620 RETURN
630 A(Q)=2:FL=1:COLOR0:X1=75:Y1=
189:W$="Misfire!! Try again.":GO
SUB120:PLAY"V25T603P4L2CL3CL8CL2
E-L8DL3DL8CL3CO2L8BO3L2CP2"
640 COLOR1:LINE(1,183)-(254,191)
,PSET,BF:LINE(111,169)-(199,177)
,PSET,BF:COLOR0:RETURN
650 PCLS:IF TI=30 THEN X1=25:Y1=
15:W$="You have run out of time.
 Sorry!":GOSUB120
660 IF RA=10 THEN X1=10:Y1=70:W$
="Good work! You had everyone ri
ght!!":GOSUB120:GOTO740 ELSEX1=3
0:Y1=25:W$="Here are the words y
ou missed.":GOSUB120
670 W=0:FORX=1TO10:S3$(X)="":S4$
(X)="":NEXTX
680 Y1=40:FORX=1TO51:IF A(X)<>2
THEN700
690 Q=X:AN$=S2$(X):GOSUB240:CA$=
S1$(X)+A$(1):X1=95:W$=CA$:GOSUB1
20:Y1=Y1+12:W=W+1:S3$(W)=S1$(X):
S4$(W)=S2$(X)
```

```
700 NEXTX
710 IN$=INKEY$:X1=50:Y1=180:W$="
Press <ENTER> to continue.":GOSU
B120
720 GOSUB1550
730 GOTO750
740 FORX=1TO5:PLAY"T30L10O3;1;2;
3;4;5;6;6;5;4;3;2;1":NEXTX
750 PCLS1:X1=70:Y1=50:W$="Your s
corecard.":GOSUB120:LINE(35,55)-
(211,150),PSET,B
760 X1=50:Y1=70:W$="NUMBER CORRE
CT :"+STR$(RA):GOSUB120
770 X1=50:Y1=90:W$="NUMBER OF QU
ESTIONS :"+STR$(NQ):GOSUB120
780 PC=INT(RA/NQ*100+.5)
790 IF PC=100 AND NQ=10 THEN PLA
Y"T10O03;1;5;9;4;2;3;6;9;6;2;7;3
;7;8;5;2;3;5;7;9;9;8;4;2;3;5;7;7
;8;7":X1=50:Y1=110:W$="CONGRATUL
ATIONS!!":GOSUB120
800 IF PC=100 AND NQ=10 THENY1=1
30 ELSE Y1=120
810 X1=50:W$="YOU SCORED"+STR$(P
C)+"%":GOSUB120
820 X1=35:Y1=170
830 IF PC=100 THENW$="PRESS <<EN
TER>>TO START AGAIN.":GOSUB120 E
LSE W$="PRESS <<ENTER>> FOR MORE
.":GOSUB120
840 IN$=INKEY$
850 GOSUB1550
860 IF PC<>100 THEN880
870 FORX=1TO51:A(X)=0:NEXTX:GOTO
70
880 PCLS1:COLOR0:FL=2:X1=80:Y1=2
0:W$="FURTHER PRACTISE.":GOSUB12
0
890 X1=40:Y1=40:W$="Which word i
s a COMPOUND word :":GOSUB120
900 LINE(40,45)-(210,45),PSET:LI
NE(90,67)-(175,130),PSET,B
910 FORZ=1TO W:AN$=S4$(Z):GOSUB2
40:CA$=S3$(Z)+A$(1):GOSUB290
920 Y1=80:FORX=1TO3:X1=100:WA$(X
)=S3$(Z)+A$(X):W$=WA$(X):GOSUB12
0:Y1=Y1+20:NEXTX
930 X1=25:Y1=175:W$="COMPOUND WO
RD":GOSUB120:LINE(110,168)-(200,
178),PSET,B
940 X1=113:Y1=175:GOSUB150
950 Y1=189
960 IF AN$=CA$ THEN X1=50:W$="GO
OD WORK! IT'S RIGHT!!":GOSUB120:
PLAY"T2V20L16O3CEGL8O4CL16O3AL4O
4CT1L1P3":GOTO990
970 X1=25:W$="SORRY, THAT'S NOT
RIGHT. TRY AGAIN.":GOSUB120:PLAY
"V25T6O3P4L2CL3CL8CL2E-L8L3DL8CL
3CO2L8BO3L2CP2T1L1P3"
```

```
980 COLOR1:LINE(1,183)-(254,191)
,PSET,BF:LINE(111,169)-(199,177)
,PSET,BF:COLOR0:GOTO940
990 PLAY"L1T1P6"
1000 IF Z=W THEN1020
1010 COLOR1:LINE(91,68)-(174,129
),PSET,BF:LINE(111,169)-(199,177
),PSET,BF:LINE(1,183)-(254,191),
PSET,BF:COLOR0
1020 NEXTZ
1030 PCLS1:X1=40:Y1=100:W$="PRES
S <<ENTER>> TO START AGAIN.":GOS
UB120:IN$=INKEY$:GOSUB1550
1040 FORX=1TO51:A(X)=0:NEXTX:GOT
O70
1050 PRINT@200,"rocket"+CHR$(128
)+"launcher";
1060 PRINT@271,"by";
1070 PRINT@331,"bob"+CHR$(128)+"
horne";
1080 A=USR0(249)
1090 PLAY"V25O2T8L4GABO3DCCEDDGF
#GDO2BGAO3CDEDCO2BABGF#GADF#AO3C
O2BABGABO3DCCEDDGF#GDO2BGABEO3DC
O2BAGDGF#L2.G"
1100 RETURN
1110 CLS:PRINT@195,"DO YOU NEED
INSTRUCTIONS ?";
1120 PRINT@261," TYPE 'Y' IF YOU
 DO,":PRINT@292," OR 'N' IF   YO
U DON'T."
1130 A=USR0(249)
1140 IN$=INKEY$
1150 IN$=INKEY$:IF IN$="N" THEN
RETURN ELSE IF IN$="Y" THEN 1160
 ELSE 1150
1160 CLS:PRINT@42,"INSTRUCTIONS.
"
1170 PRINT:PRINT" A COMPOUND WOR
D IS A WORD MADE  UP OF TWO SMAL
LER WORDS.":PRINT:PRINT" FOR EXA
MPLE 'BULL' AND 'DOZER'  COMBINE
 TO MAKE UP THE WORD     'BULLDO
ZER'."
1180 PRINT:PRINT" BULLDOZER IS A
 COMPOUND WORD.":PRINT@452,"PRES
S <ENTER> TO CONTINUE";:A=USR0(2
49):IN$=INKEY$
1190 GOSUB1550
1200 CLS:PRINT:PRINT" YOU ARE IN
 CHARGE OF A PLANET   EXPLORATIO
N ROCKET LAUNCHING    SITE."
1210 PRINT:PRINT" TOWARDS THE BO
TTOM OF THE         SCREEN, ON THE
 LEFT, IS A         SINGLE BOX CON
TAINING THE         FIRST PART OF
A WORD. TO THE      RIGHT ARE THRE
E BOXES             CONTAINING THR
EE POSSIBLE         ENDINGS."
1220 PRINT@452,"PRESS <<ENTER>>
```

```
FOR MORE.";:A=USR0(249):IN$=INKE
Y$
1230 GOSUB1550
1240 CLS:PRINT:PRINT" IF YOU MAT
CH THE TWO PARTS OF   THE WORD C
ORRECTLY, A ROCKET    WILL TAKE
OFF."
1250 PRINT:PRINT" IF YOU MAKE A
MISTAKE WHEN        TYPING THE COD
E-WORD THEN         PRESS THE BACK
-SPACE KEY (_)      AND START AGAI
N."
1260 PRINT@452,"PRESS <<ENTER>>
TO START.";:A=USR0(249):IN$=INKE
Y$
1270 GOSUB1550
1280 RETURN
1290 COLOR0,1:PCLS
1300 LINE(0,0)-(255,70),PSET,BF
1310 DRAW"S4C1":FORX=5TO230 STEP
25:DRAW"BM"+STR$(X)+",59R20DL5D2
LR2LU2L15R5D2RL2DL4D6R20U6L15":P
AINT(X+2,65):NEXTX
1320 FORX=10TO236 STEP25:PUT(X,4
3)-(X+8,57),R,PSET:NEXTX
1330 PUT(6,85)-(26,100),TV,PSET:
PUT(118,80)-(138,95),TV,PSET:PUT
(220,85)-(240,100),TV,PSET
1340 COLOR0:LINE(0,70)-(255,182)
,PSET,B
1350 FORX=30TO138 STEP12:CIRCLE(
X,76),3:NEXTX:FORX=102TO138 STEP
12:PAINT(X,76):NEXTX
1360 CIRCLE(42,86),6:PAINT(42,86
)
1370 FORX=58TO88 STEP30:LINE(X,8
0)-(X+20,84),PSET,BF:NEXTX:FORX=
58TO88 STEP30:LINE(X,86)-(X+20,9
0),PSET,BF:NEXT
1380 FORX=32TO64 STEP4:FORY=94TO
104STEP4:LINE(X,Y)-(X+4,Y+4),PSE
T,B:NEXTY,X
1390 CIRCLE(91,95),3:FORX=89TO17
8STEP9:LINE(X,99)-(X+8,107),PSET
,B:NEXTX
1400 X1=92:Y1=106:W1$="123456789
0"
1410 FORX=1TO LEN(W1$):W$=MID$(W
1$,X,1):GOSUB120:X1=X1+6:NEXTX
1420 FORY=82TO88 STEP6:FORX=148T
O180STEP8:CIRCLE(X,Y),2:PAINT(X,
Y):NEXTX,Y:CIRCLE(148,94),2:PAIN
T(148,94)
1430 LINE(158,92)-(176,96),PSET,
B
1440 FORX=146TO186 STEP18:LINE(X
,72)-(X+15,78),PSET,BF:NEXTX
1450 FORX=207TO219 STEP12:CIRCLE
(X,77),5:PAINT(X,77):NEXTX
1460 FORY=84TO92 STEP8:LINE(194,
Y)-(214,Y+4),PSET,B:NEXTY
1470 FORY=100TO104STEP4:FORX=188
TO208 STEP4:LINE(X,Y)-(X+4,Y+4),
PSET,B:NEXTX,Y
1480 X1=3:Y1=117:W$="Time":GOSUB
120:LINE(30,112)-(254,117),PSET,
BF
1490 LINE(1,110)-(254,110),PSET
1500 LINE(70,135)-(110,145),PSET
,B:FORX=120TO150 STEP15:LINE(160
,X)-(220,X+10),PSET,B:NEXTX
1510 FORX=112TO158STEP3:PSET(X,1
40):NEXTX
1520 FORY=125TO155 STEP3:PSET(11
7,Y):NEXTY:FORY=125TO155 STEP30:
FORX=117TO158STEP3:PSET(X,Y):NEX
TX,Y
1530 X1=50:Y1=175:W$="Code word"
:GOSUB120:LINE(110,168)-(200,178
),PSET,B
1540 RETURN
1550 IN$=INKEY$:IF IN$="" THEN15
50 ELSE IF IN$<>CHR$(13) THEN155
0
1560 RETURN
1570 DATA BR2UBU2U2BRBD5,BU5NDBR
3DBD4,BRUNLU3NLNURNUNRD3NLNRDBR,
BRUNLREHL2URNUR2BD4,BR3NUBL3UE3U
BL3DBD4BR3,BRHENF2HEFGBD2REBD,BU
5BRRDGBD3BR2,BR2HU3EBRBD5,BREU3H
BD5BR2,BU5BRFNLNGNENRNFBD3,BU3BR
2DNLNRDBDBR
1580 DATABRUNRDRNUDGBU2BR2,BRBU2
R2BD2,BRRULDBR2,UE3UBD5,BRHNE3U3
ERFD3GLBR2,R2U5NLD5R
1590 DATA NR3UEREUHLGBD4BR3,BUFR
EUHNLEUL3BD5BR3,BR3U5D3L3UE2RBD5
,BUFREUHL2U2R3BD5,BUNUFREUHLGU2E
RBD5BR,BU4UR3D2G3BR3,BUNUFREUHLN
GHERFGBD3BR,BRREU3HLGDFREBD3,BR2
UBU2UBD4BR
1600 DATABR2URD2GBU5URDLBD3BR,BR
2H2UE2BD5BR,BRBUNR2BU2R2BD3,BRE2
UH2BD5BR2,BR2UBUUREHL2BD5BR3,,U2
NR3U2ERFD4
1610 DATA U3NR2U2R2FGFDGL2BR3,BU
U3ERFBD3GLHBDBR3,U5RF2DG2LBR3,NR
3U3NR2U2R3BD5,U3NR2U2R3BD5,BUU3E
RBRBD3NLDGLHBDBR3,U3NU2R3NU2D3,B
RU5BD5BR2,BUFEU4NRL2BD5BR3,U5D2R
E2G2F2D,NU5R3,U5F2NDE2D5,U5F3U3D
5
1620 DATABRHU3ERFD3GLBR2,U5R2FDG
L2BD2BR3,BUU3ERFD3NHNFGLHBDBR3,U
5R2FGL2F3,BUFREUHL2UERFBD4
1630 DATA BRU5LR3BD5,NU5R3U5BD5,
BU5D3FDRUEU3BD5,NU5E2NUF2NU5,UE2
H2BR3DGFD2,BU5D2FRD2NLU2EU2BD5,N
R3UE3UL3BD5BR3,,,,,,,U2R3U2NL2D4
NL3,NU5R3U4L3BD4BR3,NR2U4R2FBD2G
BR,R3NU5U4L3D4BR3,U4R3D2L3D2R3
1640 DATABRU2NLNR2U2ERBD5,U4R3D4
NL3D2L2BE2,U4NUR2FD3,BR2U3BU2UBR
D6
1650 DATA BR2U3BU2UBD6GLBR3BU,NU
5U3NR2F3,RU5NLD5R2,U4F2NDE2D4,U4
DERFD3,U4R3D4NL3,U4R3D4L3D2BU2BR
3,U4R3D4NL3ND2,BRU4D2ERBD3,R3U2L
3U2R3BD4,BRNR2U4NR2NLUBD5BR2,NU4
R3NU4,BU4D2FDRUEU2BD4,NU4E2NUF2N
U4,UEHUBR3DGNLFD,BU4D3FR2NU4DGLB
E2,NR3UE3NL3BD4
1660 DATASCHOOL,GROUND/DROP/WAY
1670 DATARAIN,DROP/WAY/NOON
1680 DATADOOR,WAY/NOON/THING
1690 DATAAFTER,NOON/THING/GROUND
1700 DATAANY,THING/GROUND/WARDS
1710 DATAPLAY,GROUND/WARDS/WHERE
1720 DATARAIL,WAY/WARDS/THING
1730 DATAAFTER,WARDS/WHERE/BALL
1740 DATAANY,WHERE/BALL/WARDS
1750 DATABASKET,BALL/WAY/BODY
1760 DATAANY,BODY/WARDS/GROUND
1770 DATABACK,WARDS/ONE/BODY
1780 DATABATH,ROOM/ONE/WAY
1790 DATAEVERY,ONE/GROUND/WARDS
1800 DATABROAD,CAST/DOZER/NOT
1810 DATABULL,DOZER/NOT/STAIRS
1820 DATACAN,NOT/STAIRS/HOUSE
1830 DATADOWN,STAIRS/HOUSE/BODY
1840 DATALIGHT,HOUSE/PATH/MOTHER
1850 DATAEVERY,BODY/PATH/WARDS
1860 DATAFOOT,PATH/NIGHT/MOTHER
1870 DATAGRAND,MOTHER/WARDS/BODY
1880 DATAHOME,WARDS/MAN/WAY
1890 DATANO,BODY/MAN/NIGHT
1900 DATAPOLICE,MAN/DOZER/NOT
1910 DATARUN,WAY/MAN/DOZER
1920 DATAEVERY,THING/FATHER/STEA
D
1930 DATAGRAND,FATHER/STEAD/WORK
1940 DATAHOME,WORK/EVER/CREAM
1950 DATAHOW,EVER/CREAM/BALL
1960 DATAICE,CREAM/THING/WHILE
1970 DATAFOOT,BALL/WHILE/PAPER
1980 DATAMEAN,WHILE/PAPER/SELVES
1990 DATANEWS,PAPER/SELVES/HEAD
2000 DATAOUR,SELVES/HEAD/APPLE
2010 DATAOVER,HEAD/APPLE/WHERE
2020 DATAPINE,APPLE/WHERE/EVER
2030 DATAEVERY,WHERE/STAND/FALL
2040 DATAUNDER,STAND/FATHER/WORK
2050 DATAWATER,FALL/CREAM/PAPER
2060 DATASEA,SHORE/ONE/SELVES
2070 DATASOME,ONE/SELVES/SHIP
2080 DATATHEM,SELVES/ONE/SHIP
2090 DATATOWN,SHIP/SHORE/DOOR
2100 DATASEA,SHORE/DOOR/NEATH
2110 DATATRAP,DOOR/NEATH/STAIRS
2120 DATAUNDER,NEATH/PAPER/WARDS
2130 DATAUP,STAIRS/BOARD/GROUND
2140 DATAWEEK,END/SELF/THING
2150 DATAYOUR,SELF/SHORE/DOZER
2160 DATASOME,THING/SIDE/NEATH
2170 DATA189,179,237,31,152,142,
4,0,16,142,4,31,237,137,1,224,23
7,164,49,168,32,237,129,140,4,32
,38,240,57
```

SCREEN UTILITY

4K

EYE SAVER

by Norman Wong

Do your eyes ache after spending several hours using your CoCo? Well mine did, and as a result I wrote Eye Saver.

Eye Saver alters the way the CoCo shows text. Instead of dark characters on a bright background, the CoCo will use light characers on a dark background.

To use the program, first type it in and save it - do not run the program until you have saved it! Any typing errors may cause the computer to "hang" and require you to turn the computer off and on in order to regain control. In doing so, of course, you will lose the program.

Having saved a copy of the program, you may now run it. To obtain the best results, you may have to adjust the contrast and brightness controls of your television set or monitor. The only time the green background will reappear is if you CLOAD or SKIPF, but this is only temporary and can be remedied by using the CLEAR key on the Keyboard.

Eye Saver will allow you to spend more time on your CoCo.

THE LISTING:

```
10 '+--------------------+
20 '!    EYE SAVER 1.0    !
30 '! (C)1984 NORMAN WONG !
40 '+--------------------+
60 CLEAR100:S=&H1DA:PRINT"NOW PO
KING ROUTINE..."
70 POKE116,PEEK(39):POKE117,PEEK
```

```
(40)' Protect any user machine
code in high memory such as the
RAINBOW CHECK
80 SM=0:READX$,K:IFX$>""THENFORA
=1TOLEN(X$)STEP2:X=VAL("&H"+MID$
(X$,A,2)):SM=SM+X:POKES,X:S=S+1:
NEXT:IFSM<>K THENPRINT"ERROR IN"
:PRINT"   DATA"X$","K:SOUND150,1
:STOPELSE80
90 EXEC&H1DA
100 DATA8623AD9FA0029E88,957
110 DATAA11F260139DC7483,755
120 DATA009C3406318D00C3,599
130 DATA8E01678D40C3006E,756
140 DATA8E016A31A8118D35,677
150 DATAC300118E01A031A8,732
160 DATA118D2A3520303F9F,555
170 DATA749F2730889CECE4,1118
180 DATA328434069F21308D,621
190 DATA00C234103089FF64,802
200 DATAA680A7A0ACE425F8,1306
210 DATA32627EAD263406A6,709
220 DATA84A7A4867EA784EC,1258
230 DATA01ED213506ED0139,625
240 DATA0D6F266732623416,487
250 DATA9E88810D271E8108,642
260 DATA27054D2B3320218C,420
270 DATA0400234D8620E684,644
280 DATAC1602602A784A782,925
290 DATA9F88203D86208DD6,909
300 DATAD689C41F273320F4,944
310 DATA81202D2D81402508,489
320 DATA8040812025028B20,563
330 DATAA7809F888C05FF23,1025
340 DATA188E0400A68820A7,671
350 DATA808C05DF23F69F88,1072
360 DATA8620A7808C05FF23,896
370 DATAF935167E0ADB0D6F,803
380 DATA260A810C26063414,305
390 DATA8D1D35147E0AEC26,653
400 DATA0C34159EA6E61FC1,863
410 DATA9E270535157E0AFD,665
420 DATA8D053515326239C6,623
430 DATA207EA92AE01F8A35,815
440 DATA,
```

EDUCATION OVERVIEW

How Valuable Is The 'Hands-On' Lesson?

By Michael Plog, Ph.D.

When I was in high school, one project in biology class was the dissection of a frog. One girl in our class lived on a farm with a pond, and provided a host of frogs for the class. We were given little knives (scalpels), a tray half-filled with wax and other dissection equipment. We then cut the frogs apart to discover some of the mysteries of anatomical organization of living beings. This is still an exercise associated with science education.

Our teacher was a somewhat nervous and eccentric person, fighting an uphill battle trying to get her students to appreciate the workings of a biological mechanism. After I became an adult, I realized that most science laboratory teachers are somewhat nervous, and the best ones are probably a little eccentric. Every year, a student will cut a finger with a scalpel; another will faint at the realization of the mess and gore of the biological mechanism; and instead of using a straight pin to keep parts of the frog out of the way of the work area, a student will use it to draw human blood! (It is no wonder science teachers are a little nervous when turning a group of students loose with these dangerous "weapons.")

I have even more sympathy for my biology teacher, because she was also the chemistry teacher. After mixing several concoctions with difficult names, including hydrogen sulfide (smells like rotten eggs), we made gun powder. After mixing small batches of the stuff, we went outside and got to see whose batch would explode and whose batch was improperly made.

There is no doubt about it — science laboratories can be dangerous places. Acids which burn skin and clothes have to be used in chemistry classes; scalpels have to be used in biology classes. Or do they?

Imagine, if you will, the dissection of a frog without touching it — without touching a scalpel or a pin or even a wax tray. Imagine mixing volatile chemical solutions and testing for an explosion, with no fear that some student will be careless and get injured.

Such lessons are being planned right now. The frog will get dissected and the mixture will explode, but no student will get hurt. In fact, no student will be at a bench in a laboratory. The students will control the laboratory experience through a computer and video program.

The technology for this class is called computer-interactive videodiscs. That technology has been around for about 20 years, but has not been applied to middle and high school science laboratories before. The videodisc is to be connected to a computer as sort of a combination disk drive and screen controller. Students, based on keyboard entry, control which area of the videodisc is accessed and in what sequence. Thus, students can "control" the dissection of a frog by using a keyboard instead of a scalpel. All internal organs can be examined and "removed," then examined again. Students can "back up" and review the entire dissection or just one part.

A videotape would be inappropriate for this application. The videodisc is more like a "random access file," where students can skip around to different places on the disc with very little trouble. The videotape is like a "sequential access file," which has to be accessed from the beginning completely through to the end.

The amazing thing about this type of classroom use of the computer is that it is not futuristic at all. At RAINBOWfests, you can see the Color Computer connected to all sorts of video equipment. The interaction process between a videodisc and computer is certainly beyond my capabilities, but not to some of the hardware buffs who have helped enhance the Color Computer with all sorts of inventions.

The current push to implement this technology in schools is coming from the National School Board Association (NSBA). That group is promoting a $15 million public school project (called "Sci-Lab") to teach biology, physics and chemistry experiments on computer. NSBA hopes to produce 20 lessons in each of the three sciences — enough for a year in each subject. A Sci-Lab would look like a laboratory, but have work stations instead of benches. Students would face a screen and keyboard to conduct the experiment.

The price of Sci-Lab will be about the same as a new laboratory: around $100,000 for 25 work stations. A school, however, would have three laboratories in one, since each work station could conduct experiments in biology, chemistry and physics. It might even be possible to buy just a few stations or have the Sci-Lab mobile, so several districts could benefit from the same equipment.

There are, I am told, a few problems to this particular application of technology to education. (Well, there are problems to *everything*.) In the first place, the production of videodiscs is more expensive (and more complicated) than production of a videotape. Thus, you will probably not see individual school districts producing lessons on videodiscs. Instead, the lessons will be provided from NSBA itself, relying on specialized talent. After the lessons are produced, they would be given away or licensed to a marketer (depending on who financed the project).

The questions that should be asked about the techology, however, relate more to whether or not students will learn more about the subject matter from this approach or the more traditional approach. There are some positive features of this simulation approach. Students can repeat each lesson as many times as they (or their teachers) want. This may indeed help students to enhance their level of science knowledge.

NSBA representatives have also talked about a system which will allow teachers to monitor student progress and will even test students and grade exams. That would certainly free the teacher for more time for individual help to students.

What about the negative aspects of this idea? Might students learn more about controlling a keyboard than the digestive system of a frog? There is a folk wisdom used a lot by teachers that students "learn by doing." This implies an active approach to education on the part of the student.

The computer-integrated videodisc is more passive than a literal "hands-on" lesson, but more active than watching a film about a dissection. Also, if students do learn by doing, what do we want students to learn from an exercise in dissection? Not only are students supposed to learn about the insides of an animal, but are supposed to learn about processes of science as well.

Laboratory teachers have many objectives for an exercise such as dissection. Students are expected to learn about (by doing) the procedures used in a science lab. They are supposed to learn techniques that will help in more advanced scientific work. The "feel" of a scalpel through dead tissue is a skill that advanced students should have before attempting an examination of the nervous system of a mammal. That skill cannot be learned from a videodisc.

I must confess ignorance and confusion about this application of technology to education. In the first place, while it seems probable that students will learn more about the specific scientific experiment (dissection, chemical reactions, etc.) by the ability to repeat the lesson, there is no hard evidence that such additional learning will take place. There is also no hard evidence that the lack of a hands-on approach will damage students' skills or learning.

There is no guarantee that NSBA's project will ever be successful. The entire program may die from something as simple as lack of funding or something as complicated as the inability of a television screen to adequately differentiate between a nerve and a blood vessel. If successful, this application of technology to education could open entire new fields for schools as well as home education.

If you have any thoughts on this innovative use of computers in schools, I would like to hear from you. My address is 829 Evergreen, Chatham, IL 62629. We are without question living under the ancient Chinese curse: "May you live in interesting times!"

16K
Disk

LANGUAGE

Take Word Inventory with vō kab' yə ler' ē

By Keith Osani

Vocabulary is a study program suitable for all ages. It will work on a 16K or 32K computer. (If you have 16K, you must type in PCLEAR 1 before loading.) Up to 50 words may be programmed, then you can be tested on your words.

You have the option of having the computer give you the words or the definitions. If you are incorrect the computer will either ask again or go on, as you wish. After testing, the computer will give your score. You may also review your definations by quizzing the computer.

The words may be saved on your disk for future study or to be added to later with the update function.

You may also print out your vocabulary words with the DMP-200 printer or other comparable printers. The printer control codes are shown as follows:

PRINTER CONTROL CODES
FOR THE DMP-200

Line no.	Code
1510	Selects data processing mode
1520	Starts elongation
1610	End elongation
1620	Carriage return

180	218	1250	190
370	241	1430	244
580	246	1610	38
800	38	END	81
1050	16		

The listing: VOCABLRY

```
 1ø '***VOCABULARY***
 2ø '*BY KEITH OSANI*
 3ø '***2/13/85******
 4ø FOR Q=1TO1ø:R=RND(8):CLS R:NE
XT Q
 5ø PLAY"T15O1ED"
 6ø CLEAR4øøø
 7ø CLS:T=ø:QQ=ø
 8ø PRINT@1ø,"VOCABULARY":PRINT@4
ø,"BY KEITH OSANI"
 9ø PRINT@1ø6,"1) ADD WORDS
                2) GET TESTED
                3) DICTIONARY
                4) SAVE WORDS
                5) LOAD WORDS
                6) UPDATE FILE
                7) PRINT FILE"
 1øø PRINT@362,"WHICH ";:INPUT"ON
E";A
 11ø SOUND 1øø,3
 12ø IF A<1 OR A>7 THEN PRINT@394
,"(1-7)":FOR R=1TO1øøø:NEXT R:GO
TO7ø
 13ø ON A GOSUB 15ø,3øø,63ø,76ø,8
8ø,1ø4ø,148ø
 14ø GOTO6ø
 15ø CLEAR:DIM A$(5ø):DIM B$(5ø)
 16ø F=1
 17ø FOR X= F TO 1øø
 18ø CLS
 19ø PRINT@11,"add words"
 2øø PRINT"TYPE <XX> WHEN YOU ARE
 DONE"
 21ø IF X=51 THEN CLS:PRINT@23ø,"
I AM FULL":FOR B=1TO1øøø:NEXT B:
D=X-1:IF P$="UPDATE"THEN76ø ELSE
 7ø
 22ø PRINT"VOCABULARY WORD #"X
 23ø PRINT@192,"TYPE IN ";:LINE I
NPUT"WORD:";A$(X)
 24ø IF A$(X)="XX" THEN 27ø
 25ø PRINT@256,"TYPE IN ";:LINE I
NPUT"MEANING:";B$(X)
 26ø NEXT X
 27ø D=X-1:IF P$="UPDATE"THEN OO=
D:GOTO76ø
 28ø OO=D
 29ø GOTO7ø
 3øø OO=D:P=ø:CLS:PRINT@8,"vocabu
lary test":GOSUB121ø
 31ø F=ø:L=1
 32ø FOR X=1TO D:CLS
 33ø CLS:PRINT@8,"vocabulary test
"
 34ø PRINT"TYPE <XX> TO STOP TEST
ING"
 35ø PRINT@128,"";:PRINTA$(X);:LI
NE INPUT"=";S$
 36ø IF S$="XX"THEN QQ=3:X=X-1:GO
TO53ø
 37ø IF S$=B$(X) THEN 48ø
 38ø IF M=1 THEN P=P+1:GOTO33ø
 39ø FOR W=1TO1øøø:NEXT W
 4øø CLS3:PRINT@1ø6,"WRONG!!!!!";
 41ø PRINT@197,"THE CORRECT ANSWE
R IS"
 42ø PRINT@23ø,B$(X)"!!!!"
 43ø SCREENø,1
 44ø PLAY"O1T1øL3ET5L1D"
 45ø FOR T=1TO2øøø:NEXT T
 46ø IF X=D THEN 53ø
 47ø NEXT X
 48ø FOR W=1TO1øøø:NEXT W:CLS5:PR
INT@233,"CORRECT!!!":
49ø SCREENø,1
5øø F=F+1
51ø FOR Y=1TO1ø:POKE 65344,1:K=R
ND(255):L=RND(5):SOUND K,L:POKE
65344,2:Z=RND(8):CLS Z:PRINT@233
,"CORRECT!!!!!!!!!";:NEXT Y
52ø NEXT X
53ø CLS5:PRINT@5,"vocabulary tes
t ended";
54ø SCREENø,1
55ø IF QQ=3 THEN RR=D:D=X
56ø D=P+D
57ø PRINT@97,"YOU GOT "F" OUT OF
 "D:PRINT@129,"CORRECT!";
58ø FOR Y=1TO2øøø:NEXT Y
59ø D=D-P
6øø D=RR
61ø IF QQ=3 THEN D=D-1
62ø GOTO7ø
63ø CLS:PRINT@8,"dictionary mode
"
64ø D=OO
65ø PRINT"TO STOP TYPE <XX>"
66ø PRINT@224,"WHAT WOULD YOU LI
KE ":LINE INPUT"TO KNOW?";S$
67ø IF S$="XX"THEN 7ø
68ø FOR X=1TO D
69ø IF S$=A$(X) THEN 73ø
7øø IF S$=B$(X) THEN PRINT@352,S
$;:PRINT"="A$(X):GOTO74ø
71ø NEXT X
72ø PRINT@352,"I DON'T KNOW THAT
":GOTO74ø
73ø PRINT@352,S$;:PRINT"="B$(X)
74ø A$=INKEY$:IF A$=""THEN 74ø

75ø GOTO63ø
76ø CLS:PRINT@1ø,"save words"
77ø IF P$="UPDATE" THENPRINT"YOU
 MUST NAME THIS FILE ";: FOR I=2
38ø TO 2387:PRINTCHR$(PEEK(I));:
NEXT I
78ø PRINT@96,"WHAT WOULD YOU LIK
E TO":INPUT"CALL THIS FILE";S$
79ø IF LEN(S$)>8 THEN PRINT"8 LE
TTERS OR LESS PLEASE":FOR G=1TO1
øøø:NEXT G:GOTO76ø
8øø OPEN"O",#1,S$
81ø FOR X=1TO D
82ø WRITE #1,A$(X)
83ø WRITE #1,B$(X)
84ø NEXT X
85ø CLOSE#1
86ø P$=""
87ø GOTO7ø
88ø CLS:PRINT@1ø,"load words"
89ø PRINT@96,"WHAT IS THE NAME O
F":INPUT"THE FILE";S$
9øø IF LEN(S$)>8 THEN PRINT"8 LE
TTERS OR LESS PLEASE":FOR P=1TO1
øøø:NEXT P:GOTO88ø
91ø OPEN"I",#1,S$
92ø CLEAR4øøø
93ø CLEAR:DIM A$(5ø):DIM B$(5ø)
94ø X=1
95ø IF EOF(1)=-1 THEN 1øøø
96ø INPUT #1,A$(X)
97ø INPUT #1,B$(X)
98ø X=X+1
99ø GOTO95ø
1øøø CLOSE#1
1ø1ø D=X-1
1ø2ø OO=D
1ø3ø GOTO7ø
1ø4ø CLS:PRINT@1ø,"update file";
1ø5ø PRINT@128,"WHICH FILE DO YO
U WANT";:PRINT@16ø,"";:LINE INPU
T"TO UPDATE?";S$
1ø6ø IF LEN(S$)>8 THEN PRINT"8 L
ETTERS OR LESS PLEASE":FOR B=1TO
1øøø:NEXT B:GOTO1ø4ø
1ø7ø OPEN"I",#1,S$
1ø8ø CLEAR4øøø
1ø9ø DIM A$(5ø):DIM B$(5ø)
11øø X=1
111ø IF EOF(1)=-1 THEN116ø
112ø INPUT #1,A$(X)
113ø INPUT #1,B$(X)
114ø X=X+1
115ø GOTO111ø
116ø CLOSE#1
117ø F=X
118ø P$="UPDATE"
119ø OO=D
12øø GOTO17ø
121ø PRINT@128,"WOULD YOU LIKE M
E TO              1>GIVE YOU THE W
ORD                 AND YOU GIVE T
HE                  DEFINITION
                  2>GIVE YOU THE D
EF-                 INITION AND YO
U                   GIVE ME THE WO
RD"
122ø PRINT@384,"";:INPUT"WELL";A
123ø IF A<1 OR A>2 THEN 122ø
124ø GOSUB144ø
125ø IF A=1 THEN CLS:PRINT@8,"vo
cabulary test":RETURN
126ø CLS:F=ø:P=ø:L=1
127ø FOR X=1 TO D:CLS
128ø CLS:PRINT@8,"vocabulary tes
t"
129ø PRINT"TYPE <XX> TO STOP TES
TING"
13øø PRINT@128,B$(X);:LINE INPUT
"=";S$
131ø IF S$="XX" THEN QQ=3:X=X-1:
GOTO53ø
132ø IF S$=A$(X) THEN 14øø
133ø IF M=1 THEN P=P+1:GOTO128ø
134ø CLS3:PRINT@1ø6,"WRONG!!!!!
";
135ø PRINT@197,"THE CORRECT ANSW
ER IS":PRINT@23ø,A$(X)"!!!!"
136ø SCREENø,1:PLAY"O1T1øL3ET5L1
D"
137ø FOR A=1TO1øøø:NEXT A
138ø NEXT X
139ø GOTO53ø
14øø CLS5:PRINT@233,"CORRECT!!!"
;:SCREENø,1
141ø F=F+1:FOR Y=1TO1ø:POKE65344
,1:K=RND(255):L=RND(5):SOUND K,L
:POKE 65344,2:Z=RND(8):CLS Z:PRI
NT@233,"CORRECT!!!!!";:NEXT Y
142ø NEXT X
143ø GOTO53ø
144ø CLS:PRINT@224,"WOULD YOU LI
KE ME TO              1>GIVE YOU T
HE QUESTION AGAIN       IF YOU GET
 IT WRONG              2>JUST TELL
 YOU THE RIGHT            ANSWER AN
D GO ON TESTING."
145ø PRINT@384,"WE";:INPUT"LL";M
146ø IF M>2 OR M<1 THEN 144ø
147ø RETURN
148ø CLS:PRINT@1ø,"print file"
149ø I=(PEEK(65314)AND1)
15øø IF I=1 THEN PRINT"TURN ON Y
OUR PRINTER":O=9
151ø PRINT#-2,CHR$(19)
152ø PRINT#-2,CHR$(27);CHR$(14)
153ø I=(PEEK(65314) AND 1)
154ø IF I=ø AND O=9 THEN PRINT"T
HANKS!!"
155ø O=ø
156ø PRINT@128,"WHICH FILE WOULD
 YOU":LINE INPUT"LIKE TO PRINT O
UT?";R$
157ø IF LEN(R$)>8 THEN PRINT"8 L
ETTERS OR LESS PLEASE":FOR P=1TO
1øøø:NEXT P:GOTO148ø
158ø R=LEN(R$):Z=21-R/2:PRINT#-2
,TAB(Z)
159ø PRINT@23ø,"printing......"
16øø PRINT#-2,R$:PRINT#-2,TAB(16
):PRINT#-2,"VOCABULARY"
161ø PRINT#-2,CHR$(27);CHR$(15)
162ø FOR T=1TO 2:PRINT#-2,CHR$(1
3):NEXT T
163ø OPEN"I",#1,R$
164ø CLEAR4øøø
165ø DIM A$(5ø):DIM B$(5ø)
166ø X=1
167ø IF EOF(1)=-1 THEN 173ø
168ø INPUT#1,A$(X)
169ø INPUT#1,B$(X)
17øø PRINT#-2,X;A$(X)"="B$(X)
171ø X=X+1
172ø GOTO167ø
173ø CLOSE #1:X=ø:GOTO6ø
```

LEARN THOSE 'TRICKY WORDS'

By Penny L. Rand

This program provides practice in the correct use of frequently misused contractions and possessives (its, it's, your, you're, etc.). It works well with my seventh graders; language arts teachers from grades three on up should find it useful. It can be easily modified to include vocabulary and spelling words for any grade level.

Tricky Words uses a randomly colored background and a colorful border to highlight a sentence which is missing a word. Two choices are given, one of which will correctly fill in the blank. I designated the choices '1' and '2' instead of 'A' and 'B' because numbers are easier to find and closer together on the keyboard than letters.

If the correct answer is picked, one of 10 different words indicating success will flash on the screen. If an incorrect response is given, one of 10 different words indicating an error will flash, then the correct answer will be printed on the screen. If the same response is always given when the student picks the right or wrong answer, it can quickly become boring, so I wanted to make sure each response is different.

If the student completes eight out of 10 questions correctly, he will be greeted by a message which says he has done a good job and is finished. If, however, the student gets less than eight correct, he will be notified by a message that he must do the exercise again until he gets at least eight correct; the computer will automatically restart the program.

Tricky Words can be easily modified for use with any kind of exercise which can be set up to have two choices of answers, one of which correctly fits into a blank in a sentence or correctly answers a short question. Changing the sentences or questions only involves retyping the last 10 data statements of the program, lines 401 through 410. Each statement contains seven pieces of information:

1) First, type the sentence or question. Put in the appropriate end mark for the sentence and follow it by a comma. Try to keep the sentence to 29 spaces or less. If more than 29 spaces are used, the program will still work, but the sentence will bleed into the border and will not look as attractive.

2) Type one of the two choices of answers you wish to provide for the sentence. Follow this with a comma.

3) Type the other answer choice you wish to provide. Follow this with a comma.

4) Type either 1 or 2, whichever is the placement of the correct answer ('1' if the correct answer is the first of the two choices listed; '2' if you listed the correct choice second). Follow this number with a comma.

5) Type the correct answer again. (This is the data the computer will use to inform the student of the correct answer if he chose the wrong answer to begin with.) Type a comma.

6) Type whatever word you wish to use to indicate success ("great," "good," "excellent," etc.). Type in a comma.

7) Type whatever word you want to indicate an error ("sorry," "wrong," "too bad," etc.). Do *not* type in a comma this time.

The listing: TRIKWORD

120 50
250 230
END 215

```
1Ø 'TRICKY WORDS
11 'PENNY RAND
15 'R2 BOX 385E
2Ø 'PRESQUE ISLE, MAINE Ø4769
25 'JANUARY, 1985
35 CLS3
4Ø GOSUB3ØØ
8Ø PRINT @234,"TRICKY WORDS";
85 FOR X=1 TO 2ØØØ:NEXT X
9Ø CLS4
91 PRINT@229,"TYPE YOUR FIRST NA
ME";:PRINT@296,"AND PUSH ENTER."
;
92 GOSUB3ØØ
93 INPUTN$
94 C=Ø:W=Ø:RESTORE
95 FOR M= 1 TO 1Ø
96 Y=RND(8):IFY=1 THEN 96
97 CLS(Y)
99 PRINT@65,"PRESS 1 OR 2 TO IND
ICATE":PRINT@97,"THE CORRECT ANS
WER. THEN":PRINT@129,"wait FOR T
HE NEXT QUESTION."
1ØØ GOSUB3ØØ
11Ø READ A1$,A2$,A3$,A4$,A5$,A6$
,A7$
119 'SET UP QUESTION AND TWO ANS
WERS
12Ø PRINT@194,A1$;:PRINT@265,"1)
"A2$;:PRINT@297,"2)"A3$;
13Ø B$=INKEY$
14Ø IFB$=""THEN 13Ø
149 'CORRECT RESPONSE
15Ø IFB$=A4$THEN PRINT@386,A6$"!
";:SOUND 147,5:SOUND 175,5:FOR T
=1 TO 5ØØ:NEXT T:C=C+1
159 'INCORRECT RESPONSE
16Ø IFB$<>A4$THEN SOUND 5Ø,5:PRI
NT@353,A7$", "N$".";:PRINT@385,"
THE CORRECT ANSWER IS "A5$".";:F
OR T=1 TO 18ØØ:NEXT T
17Ø NEXT M
2ØØ CLS(Y)
2Ø5 IFY=1 THEN 2ØØ
21Ø PRINT@65,N$",";
219 'END DISPLAY IF SATISFACTORI
LY COMPLETED
22Ø IF C>7 THEN PRINT@161,"YOU G
OT"C"CORRECT OUT OF 1Ø.":PRINT@1
93,"YOU ARE FINISHED.  GOOD JOB!
":GOSUB3ØØ
229 'END DISPLAY IF UNSATISFACTO
RILY COMPLETED
23Ø IF C<8THEN PRINT@161,"YOU GO
T"C"CORRECT OUT OF 1Ø.":PRINT@19
3,"YOU MUST DO THIS PROGRAM OVER
":PRINT@225,"UNTIL YOU GET AT LE
AST 8 RIGHT.":PRINT@289,"wait!!
THE COMPUTER WILL RESET":PRINT@3
21,"ITSELF.":GOSUB3ØØ:FOR X=1 TO
 7ØØØ:NEXT X:GOTO94
25Ø GOTO25Ø
29Ø 'BORDER
3ØØ H=RND(128)+127
31Ø FOR X=32 TO 63:PRINT@X,CHR$(
H);:NEXT X
32Ø FOR X=48Ø TO 511:PRINT@X,CHR
$(H);:NEXT X
33Ø FOR X=Ø TO 448 STEP 32:PRINT
@X,CHR$(H);:NEXT X
34Ø FOR X=31 TO 479 STEP 32:PRIN
T@X,CHR$(H);:NEXT X
35Ø RETURN
4ØØ 'QUESTIONS AND ANSWERS
4Ø1 DATA ?  HIS APPLE.,ITS ,IT'S
,2,IT'S,CORRECT,SORRY
4Ø2 DATA ?  MAIL IS LATE.,YOUR
,YOU'RE,1,YOUR,GOOD GOING,WHOOPS
4Ø3 DATA ?  BOOKS ARE RUINED.,TH
EIR,THERE,1,THEIR,ALL RIGHT,TOO
BAD
4Ø4 DATA LET'S GO  ?  THE MOVIES
.,TOO,TO ,2,TO,GREAT,AFRAID NOT
4Ø5 DATA ?  ARE SIX BOYS ABSENT.
,THERE  ,THEY'RF,1,THERE,NICE GO
ING,NOT QUITE
4Ø6 DATA THE CAR HAS LOST  ?  SH
INE.,ITS ,IT'S,1,ITS,EXCELLENT,I
NCORRECT
4Ø7 DATA I EARNED  ?  DOLLARS.,T
OO,TWO,2,TWO,SUPER,NOT EXACTLY
4Ø8 DATA ?  NOT BEING QUIET!,THE
Y'RE,THEIR  ,1,THEY'RE,VERY NICE
,NOT REALLY
4Ø9 DATA I LIKE CHOCOLATE  ?  MU
CH.,TO ,TOO,2,TOO,WAY TO GO,WRON
G ANSWER
41Ø DATA ?  INVITED TO MY PARTY.
,YOUR  ,YOU'RE,2,YOU'RE,GOOD,OQP
S
```

32K
ECB

How To Be An Educated Stargazer In 12 Easy Lessons

By Eugene Vasconi

Have you ever stood outside under the night sky staring into the heavens while trying to remember what your science teacher said about finding constellations? Then you get frustrated because you can't even find the "Big Dipper."

STARFINDER is designed to accurately display 12 of the more picturesque constellations in the night sky. In addition, the data option provides information on each constellation, including alternate names, selected member stars and comments on the mythological figures ancients used to describe them. "Scorpious", for example, was depicted as a scorpion and both claws and tail are labeled. Finally, a LINE command is available to connect each star and make the figure more apparent.

Selecting number 13 on the menu, Dippers, displays the Big and Little Dippers on the star field for use in visualizing their relative positions. Press "D" and the computer will show youhow the two are related by distance and angle.

In assembling star coordinates, I have maintained as great an accuracy with member stars as is possible given the graphics area available. Size is the only variable and one constellation versus another might no be an accurate comparison. "Pegasus,"for example, displayed much clearer as a tight figure than spread out. However, when using STARFINDER to help you locate actual constellations, you will find it quite realistic.

Actual constellations are not necessarily made up of the brightest stars. I have included an intensity option which allows you to choose a more realistic desplay of somewhat dim member stars or one that would make them stand out. Selecting this option at the beginning will cause all stars displayed to be in that mode.

A large part of the program is made up of Hi-Res labels and it is worth mentioning the technique. I have elaborated on a system Mike Hall used in his TRAILIN' TAIL program (RAINBOW, August 1983) which displays Hi-Res words by detecting their ASCII equivalent. (I heartily recommend this way if you fnd a program needs a lot of wording on the graphics screen.) In STARFINDER this system consists of lines 3, 110, 111, 113-149, plus individual draw coordinates and GOSUBs for the particular label.

STARFINDER is not a game but a learning tool for both students and those of us who have forgotten our planetarium visits. I think a few sessions with the program will make our star-gazing much more enjoyable.

A final comment for those who might be curious about the connect sequence: I discovered that while DATA statements were fine for the display of the stars alone, this was unsatisfactory for connecting them with lines since each constellation has its own peculiar design. The easiest solution was to create a central universal line routine that could be joined in progress at various positions according to how many stars were involved, then, by assigning the variables array designations, the lines would draw properly. Loops were set up tto take care of as many lines as possible (lines 49-57).

12	209	106	196
24	187	118	230
44	87	148	190
60	201	163	89
71	191	177	3
83	56	END	150
93	248		

The listing: STARFIND

```
1 CLEAR2ØØ
2 ' **TITLE SCREEN **
3 DIM LT$(43):FORX=1TO43:READ LT
$(X):NEXTX:DIMST(165),SR(165)
4 PMODE4:PCLS:SCREEN1,1:DRAW"BM1
1,3Ø;C1;S16":N$="STARFINDER":GOS
UB11Ø
5 FORTP=1TO15:PLAY"V31":FORTQ=1T
O4:R=RND(255):S=RND(196):PLAY"L1
1T31V<;3;4;5;6;7":PSET(R,S):NEXT
TQ,TP
6 FORTC=1TO3Ø:PLAY"T25ØL25ØV15O1
CG":CIRCLE(125,9Ø),TC:NEXTTC
7 DRAW"BM8Ø,15Ø;C1;S4":N$="BY EU
GENE VASCONI":GOSUB11Ø
8 FORTX=45TO6ØSTEP2:CIRCLE(125,9
Ø),TX,,.3,.83,.68:NEXTTX
```

```
9 DRAW"BM1Ø,19Ø;C1;S4":N$="LOADI
NG":GOSUB11Ø
1Ø FORTL=1TO165:READST(TL),SR(TL
):PLAY"L54T18O4;1;8":NEXTTL
11 DRAW"BM1Ø,19Ø;CØ":N$="LOADING
":GOSUB11Ø:DRAW"BM1Ø,19Ø;C1":N$=
"PRESS ANY KEY TO BEGIN":GOSUB11
Ø
12 CIRCLE(125,9Ø),6Ø,1,.3,.83,.6
8:IN$=INKEY$:CIRCLE(125,9Ø),6Ø,Ø
,.3,.83,.68:PLAY"L255T255O5GG#":
IF IN$=""THEN12
13 ' ** INTENSITY SELECTION **
14 CLS(Ø):PRINT@99,"STAR INTENSI
TY SELECTION":PRINT:PRINT"(1) WI
LL CREATE MORE REALISTIC  STAR P
ATTERNS":PRINT:PRINT"(2) WILL EM
PHASIZE STARS":PRINT:INPUT"SELEC
T (1) OR (2)";SS:IF SS<1 OR SS>2
 THEN14
15 CLS:SOUND2ØØ,1 '**MENU PAGE*
16 PRINT@5,"]]] constellations [
[[":PRINT:PRINT"(1)  TAURUS":PRI
NT"(2)  ORION":PRINT"(3)  URSA M
INOR":PRINT"(4)  GEMINI":PRINT"(
5)  CANUS MAJOR":PRINT"(6)  LEO"
:PRINT"(7)  URSA MAJOR"
17 PRINT"(8)  SCORPIUS":PRINT"(9
)  CYGNUS":PRINT"(1Ø) PEGASUS":P
RINT"(11) PISCES":PRINT"(12) CAS
SIOPEIA":PRINT"(13) THE DIPPERS"
18 INPUT"                 select
ion ";SL:PLAY"V31":FORLP=1TO4:PL
AY"L11T31V<;O5;3;4;5;6;7;8;9;1Ø"
:NEXTLP:IF SL<1OR SL>13 THEN15
19 PLAY"V15":ON SL GOSUB28,29,3Ø
,31,32,33,34,35,36,37,38,39,4Ø
2Ø '**** DISPLAY SEQUENCE******
21 PMODE4:PCLS:SCREEN1,1:
22 FORY=1TO15Ø:R=RND(255):S=RND(
196):PSET(R,S):NEXTY
23 FORCC=W TO Z:CIRCLE(ST(CC),SR
(CC)),SS:PLAY"L54T18O4;1;8;1;8;1
;8":NEXTCC
24 DRAW"BM1Ø,19Ø;S4;C1":N$="M FO
R MENU OR D FOR DATA":GOSUB11Ø
25 RT$=INKEY$:IF RT$="M" THEN15E
LSE26
26 IF RT$="D"THEN42ELSE25
27 'VARIABLES
28 W=1:Z=12:CD=1:RETURN
29 W=13:Z=26:CD=2:RETURN
3Ø W=27:Z=33:CD=3:RETURN
31 W=34:Z=49:CD=4:RETURN
32 W=5Ø:Z=63:CD=5:RETURN
33 W=64:Z=78:CD=6:RETURN
34 W=79:Z=94:CD=7:RETURN
35 W=95:Z=11Ø:CD=8:RETURN
36 W=111:Z=118:CD=9:RETURN
37 W=119:Z=131:CD=1Ø:RETURN
```

```
38 W=132:Z=146:CD=11:RETURN
39 W=147:Z=151:CD=12:RETURN
4Ø W=152:Z=165:CD=13:RETURN
41 ' ** DATA SELECTOR **
42 PLAY"L1ØØT8O4AB-O2B-BO1C#D":D
RAW"BM1Ø,19Ø;S4;CØ":N$="M FOR ME
NU OR D FOR DATA":GOSUB11Ø
43 PLAY"L6T4ØO2;1;8;1;8;1;8":ON
CD GOSUB61,64,68,71,74,77,81,84,
87,91,95,99,1Ø3
44 DRAW"BM1Ø,19Ø;S4;C1":N$="M FO
R MENU   L TO CONNECT":GOSUB11Ø
45 RT$=INKEY$:IF RT$="M" THEN15E
LSE46
46 IF RT$="L" THEN47ELSE45
47 PLAY"L255O3GG#O4GG#O5GG#":ON
CD GOTO178,18Ø,182,184,186,189,1
91,194,196,198,2ØØ,2Ø3,2Ø5
48 ' **CONNECTION SEQUENCE**
49 LINE(ST(M),SR(M))-(ST(N),SR(N
)),PSET:LINE-(ST(P),SR(P)),PSET
5Ø LINE(ST(Q),SR(Q))-(ST(R),SR(R
)),PSET:LINE-(ST(S),SR(S)),PSET
51 IF SK=3THEN56
52 LINE(ST(T),SR(T))-(ST(U),SR(U
)),PSET
53 IF SK=1THEN 57
54 LINE(ST(V),SR(V))-(ST(W),SR(W
)),PSET
55 IF SK=2THEN57
56 FORLP=GG TO HH:PLAY"L255O4B-B
C":LINE(ST(LP),SR(LP))-(ST(LP+1)
,SR(LP+1)),PSET:NEXTLP
57 FORLP=G TO H:PLAY"L255O4B-BC"
:LINE(ST(LP),SR(LP))-(ST(LP+1),S
R(LP+1)),PSET:NEXTLP
58 DRAW"BM1Ø,19Ø;S4;CØ":N$="M FO
R MENU   L TO CONNECT":GOSUB11Ø:
DRAW"BM1Ø,19Ø;C1":N$="M FOR MENU
":GOSUB11Ø
59 A$=INKEY$:IF A$="M" THEN 15 E
LSE 59
6Ø '**STAR DATA**
61 DRAW"BM1Ø5,3Ø;S16;C1":N$="TAU
RUS":GOSUB11Ø:DRAW"BM199,45;S4":
N$="THE BULL":GOSUB11Ø
62 DRAW"BM43,96;S4":N$="ALDEBARA
N":GOSUB11Ø:DRAW"BM14,35;S4;C1":
N$="HORNS":GOSUB11Ø:DRAW"BM176,1
4Ø;S4;C1":N$="LEGS":GOSUB11Ø
63 RETURN
64 DRAW"BM15,28;S16;C1":N$="ORIO
N":GOSUB11Ø:DRAW"BM135,1Ø;S4":N$
="THE HUNTER":GOSUB11Ø
65 DRAW"BM159,138;S4":N$="RIGEL"
:GOSUB11Ø:DRAW"BM19Ø,63":N$="ARM
":GOSUB11Ø
66 DRAW"BM82,1Ø9;S4;C1":N$="BELT
":GOSUB11Ø:DRAW"BM66,52":N$="ARM
":GOSUB11Ø
```

```
67 RETURN
68 DRAW"BM125,2Ø;S8;C1":N$="URSA
 MINOR":GOSUB11Ø:DRAW"BM17Ø,32;S
4":N$="LITTLE BEAR":GOSUB11Ø:DRA
W"BM158,42":N$="LITTLE DIPPER":G
OSUB11Ø
69 DRAW"BM51,52;S4;C1":N$="POLAR
IS":GOSUB11Ø:DRAW"BM188,117":N$=
"HEAD":GOSUB11Ø:DRAW"BM4Ø,1ØØ;S4
":N$="TAIL":GOSUB11Ø
7Ø RETURN
71 DRAW"BM19,135;S8;C1":N$="GEMI
NI":GOSUB11Ø:DRAW"BM27,15Ø;S4":N
$="THE TWINS":GOSUB11Ø
72 DRAW"BM6,43;S4":N$="POLLUX":G
OSUB11Ø:DRAW"BM32,16":N$="CASTOR
":GOSUB11Ø
73 RETURN
74 DRAW"BM11,2Ø;S8;C1":N$="CANUS
":GOSUB11Ø:DRAW"BM11,35":N$="MAJ
OR":GOSUB11Ø
75 DRAW"BM12,47;S4":N$="GREAT DO
G":GOSUB11Ø:DRAW"BM149,43;S4":N$
="SIRIUS":GOSUB11Ø:DRAW"BM197,66
":N$="MIRZAM":GOSUB11Ø
76 RETURN
77 DRAW"BM11,3Ø;S16;C1":N$="LEO"
:GOSUB11Ø:DRAW"BM84,18;S8":N$="T
HE LION":GOSUB11Ø
78 DRAW"BM157,88;S4;C1":N$="ALGI
EBA":GOSUB11Ø:DRAW"BM177,14Ø":N$
="REGULUS":GOSUB11Ø
79 DRAW"BM2Ø4,55":N$="HEAD":GOSU
B11Ø:DRAW"BM14,115":N$="TAIL":GO
SUB11Ø
8Ø RETURN
81 DRAW"BM9,2Ø;S8;C1":N$="URSA M
AJOR":GOSUB11Ø
82 DRAW"BM1Ø,3Ø;S4":N$="GREAT BE
AR":GOSUB11Ø:DRAW"BM26,1ØØ;S4":N
$="BIG DIPPER":GOSUB11Ø:DRAW"BM1
53,1ØØ":N$="MERAK":GOSUB11Ø:DRAW
"BM1Ø9,65":N$="DUBHE":GOSUB11Ø
83 RETURN
84 DRAW"BM9,2Ø;S8;C1":N$="SCORPI
US":GOSUB11Ø
85 DRAW"BM1Ø1,54;S4":N$="ANTARES
":GOSUB11Ø:DRAW"BM25,115":N$="SC
HAULA":GOSUB11Ø:DRAW"BM2ØØ,52":N
$="CLAWS":GOSUB11Ø:DRAW"BM26,15Ø
":N$="TAIL":GOSUB11Ø
86 RETURN
87 DRAW"BM9,15;S8;C1":N$="CYGNUS
":GOSUB11Ø
88 DRAW"BM1Ø,25;S4":N$="THE SWAN
":GOSUB11Ø:DRAW"BM1Ø,35":N$="NOR
THERN CROSS":GOSUB11Ø
89 DRAW"BM1Ø7,53":N$="DENEB":GOS
UB11Ø:DRAW"BM127,87":N$="SADR":G
OSUB11Ø:DRAW"BM186,166":N$="HEAD
":GOSUB11Ø:DRAW"BM19Ø,5Ø":N$="WI
NG":GOSUB11Ø:DRAW"BM44,135":N$="
WING":GOSUB11Ø
9Ø RETURN
91 DRAW"BM9,2Ø;S8;C1":N$="PEGASU
S":GOSUB11Ø:DRAW"BM1Ø4,15;S4":N$
="WINGED HORSE":GOSUB11Ø
92 DRAW"BM119,66":N$="MARKAB":GO
SUB11Ø:DRAW"BM117,12Ø":N$="SCHEA
T":GOSUB11Ø:DRAW"BM171,82":N$="A
LGENIB":GOSUB11Ø
93 DRAW"BM44,59":N$="HEAD":GOSUB
11Ø:DRAW"BM16,125":N$="FRONT LEG
S":GOSUB11Ø
94 RETURN
95 DRAW"BM9,2Ø;S8;C1":N$="PISCES
":GOSUB11Ø:DRAW"BM9Ø,15;S4":N$="
THE FISHES":GOSUB11Ø
96 DRAW"BM27,115;S4":N$="ALRISCH
A":GOSUB11Ø:DRAW"BM126,58":N$="F
ISH":GOSUB11Ø:DRAW"BM19Ø,115":N$
="FISH":GOSUB11Ø
97 DRAW"BM4Ø,155":N$="FISH ARE C
ONNECTED BY A ROPE":GOSUB11Ø
98 RETURN
99 DRAW"BM9,3Ø;S16;C1":N$="CASSI
OPEIA":GOSUB11Ø
1ØØ DRAW"BM189,1Ø8;S4":N$="CAPH"
:GOSUB11Ø:DRAW"BM151,137":N$="SC
HEDIR":GOSUB11Ø
1Ø1 DRAW"BM138,11Ø":N$="TORSO":G
OSUB11Ø:DRAW"BM8,1ØØ":N$="KNEE O
F WOMAN":GOSUB11Ø:DRAW"BM22,11Ø"
:N$="ON THRONE":GOSUB11Ø
1Ø2 RETURN
1Ø3 DRAW"BM1Ø,1Ø4;S4;C1":N$="LIT
TLE DIPPER":GOSUB11Ø:DRAW"BM158,
14Ø":N$="BIG DIPPER":GOSUB11Ø
1Ø4 DRAW"BM39,28;":N$="POLARIS":
GOSUB11Ø:DRAW"BM192,6Ø":N$="POIN
TERS":GOSUB11Ø
1Ø5 FOREX=1TO2Ø:PLAY"L255O2CC#":
LINE(18Ø,74)-(2ØØ,62),PSET:LINE-
(196,8Ø),PSET:FOREY=1TO5Ø:NEXTEY
:COLOR1:LINE(18Ø,74)-(2ØØ,62),PS
ET:LINE-(196,8Ø),PSET:COLORØ:NEX
TEX
1Ø6 FOREX=1TO15:LINE(174,76)-(91
,28),PSET:COLORØ:FOREY=1TO5Ø:NEX
TEY:LINE(174,76)-(91,28),PSET:CO
LOR1:PLAY"L25ØT25ØV-O4;1;3;1;3":
NEXTEX:PLAY"V15"
1Ø7 LINE(93,25)-(155,25),PSET:LI
NE-(173,73),PSET:PLAY"L4ØT2O5ABA
":DRAW"BM156,1Ø;S4;C1":N$="5 TIM
ES":GOSUB11Ø:DRAW"BM14Ø,2Ø":N$="
POINTER DISTANCE":GOSUB11Ø
1Ø8 RETURN
1Ø9 ' **LABEL SUBROUTINE**
11Ø L=LEN(N$):FORZ=1TOL:M=ASC(MI
```

```
D$(N$,Z,1))-47:IFM=-15THENDRAW"B
R4"ELSEDRAWLT$(M)
111 PLAY"L255T255O5V15BC":DRAW"B
R2":NEXT:RETURN
112 ' **LETTER DATA**
113 DATABRHU4ERFD4GNLBR2
114 DATAR2U6NGD6R2
115 DATABU5ER2FDGL2GD2R4
116 DATABU5ER2FDGNLFDGL2NHBR3
117 DATABR3U6G3R4BD3
118 DATABUFR2EU2HL3U2R4BD6
119 DATABU3R3FDGL2HU4ER2BD6BR
12Ø DATABU6R4DG3D2BR3
121 DATABRHUER2EUHL2GDFR2FDGNL2B
R
122 DATABRR2EU4HL2GDFR3BD3
123 DATABR4,BR4,BR4,BR4,BR4,BR4,
BR4
124 DATAU5ER2FD2NL4D3
125 DATARU6NLR2FDGNL2FDGNL3BR
126 DATABR4BU5HL2GD4FR2EBD
127 DATARU6NLR2FD4GNL2BR
128 DATAU6NR4D3NR3D3R4
129 DATAU3NR3U3R4BD6
13Ø DATABUU4ER3BD4NLD2L3NHR3
131 DATAU3NU3R4NU3D3
132 DATAR2U6NL2NR2D6R2
133 DATABUNUFR2ENU5BD
134 DATAU3NU3RNE3F3
135 DATANU6R4
136 DATAU6F2DUE2D6
137 DATAU6F4NU4D2
138 DATABRHU4ER2FD4GNL2BR
139 DATAU6R3FDGL3D3BR4
14Ø DATABRHU4ER2FD4GNL2BUHF2
141 DATAU6R3FDGL3RF3
142 DATABUFR2EUHL2HUER2FBD5
143 DATABU6R4L2D6BR2
144 DATABUNU5FR2ENU5BD
145 DATABU6D4F2E2U4BD6
146 DATANU6E2UDF2NU6
147 DATAUE4NUG2H2NUF4D
148 DATABU6DF2E2NUG2D3BR2
149 DATABU6R4DG4DR4
15Ø ' **STAR POSITIONS**
151 'TAURUS
152 DATA4Ø,52,96,84,112,84,14Ø,1
ØØ,184,96,1Ø8,76,1Ø2,68,52,2Ø,19
2,112,144,136,12Ø,12Ø,1ØØ,112
153 'ORION
154 DATA176,36,183,56,184,64,183
,68,18Ø,84,14Ø,68,128,52,1ØØ,6Ø,
116,1Ø4,124,1ØØ,132,95,152,132,1
Ø8,14Ø,88,48
155 'URSA MINOR
156 DATA43,48,61,78,89,1Ø7,134,1
19,135,142,18Ø,14Ø,179,114
157 'GEMINI
158 DATA44,49,62,61,92,1Ø5,124,1
18,183,156,164,186,77,52,42,8Ø,7
Ø,18,82,2Ø,115,36,168,8Ø,232,1Ø2
,2Ø1,122,152,6,84,54
159 'CANUS MAJOR
16Ø DATA12Ø,18Ø,1Ø4,148,98,138,8
6,125,118,1Ø8,157,7Ø,141,45,124,
6,1Ø2,34,98,1Ø4,19Ø,6Ø,182,166,7
3,128,53,15Ø
161 'LEO
162 DATA14,15Ø,66,1Ø6,84,1Ø4,15Ø
,66,192,34,2Ø3,44,228,41,231,18,
59,172,73,134,184,125,173,1ØØ,24
4,13Ø,149,88,145,16Ø
163 'URSA MAJOR
164 DATA15,95,44,8Ø,7Ø,84,96,84,
11Ø,1Ø1,149,9Ø,146,67,198,43,236
,37,197,63,224,89,248,9Ø,249,98,
156,149,2Ø4,176,2Ø7,16Ø
165 'SCORPIUS
166 DATA74,12Ø,61,133,54,141,7Ø,
161,98,156,114,144,114,118,116,1
ØØ,138,66,148,56,191,19,195,36,1
92,6Ø,188,75,162,54,69,12Ø
167 'CYGNUS
168 DATA99,47,114,79,139,1Ø8,173
,16Ø,182,23,163,59,74,1Ø6,32,118
169 'PEGASUS
17Ø DATA83,125,1Ø3,114,115,1Ø8,1
57,11Ø,164,72,114,71,1Ø1,66,96,6
3,8Ø,57,64,1Ø6,76,1Ø3,1Ø3,95,1Ø5
,98
171 'PISCES
172 DATA81,1Ø7,92,91,1ØØ,78,1Ø4,
69,11Ø,58,111,46,1Ø4,54,116,95,1
25,96,154,96,167,98,175,95,183,1
Ø2,178,1Ø7,167,1Ø6
173 'CASSIOPEIA
174 DATA66,61,91,95,124,93,144,1
29,181,1Ø2
175 'DIPPERS
176 DATA89,26,8Ø,38,73,52,76,68,
67,74,78,89,84,82,133,162,138,13
8,151,126,166,11Ø,183,112,194,85
,176,76
177 '***CONNECT INFO***
178 M=9:N=5:P=1Ø:Q=11:R=4:S=12:T
=3:U=6:SK=1:G=1:H=7
179 GOTO49
```

```
18Ø Q=25:R=21:S=2Ø:T=2Ø:U=26:V=2
3:W=18:SK=2:G=13:H=23
181 GOTO5Ø
182 V=3Ø:W=33:G=27:H=32:SK=2
183 GOTO54
184 M=4Ø:N=35:P=41:Q=48:R=43:S=4
9:T=39:U=36:V=47:W=45:G=34:H=37:
GG=42:HH=45:SK=Ø
185 GOTO49
186 LINE(ST(58),SR(58))-(ST(56),
SR(56)),PSET:LINE-(ST(59),SR(59)
),PSET
187 M=59:N=53:P=61:Q=53:R=62:S=6
3:T=6Ø:U=55:SK=1:G=5Ø:H=57
188 GOTO49
189 M=75:N=77:P=67:Q=65:R=73:S=7
8:G=64:H=7Ø:GG=72:HH=75:SK=3
19Ø GOTO49
191 LINE(ST(85),SR(85))-(ST(82),
SR(82)),PSET:LINE(ST(83),SR(83))
-(ST(92),SR(92)),PSET
192 M=94:N=92:P=93:Q=86:R=88:S=8
9:T=89:U=91:V=84:W=88:G=79:H=84:
GG=85:HH=89:SK=Ø
193 GOTO49
194 Q=1Ø7:R=1Ø8:S=1Ø4:T=1Ø4:U=1Ø
9:G=95:H=1Ø5:SK=1
195 GOTO5Ø
196 Q=115:R=116:S=112:T=112:U=11
7:V=117:W=118:G=111:H=113:SK=2
197 GOTO5Ø
198 Q=131:R=121:S=124:G=128:H=13
Ø:GG=119:HH=126:SK=3
199 GOTO5Ø
2ØØ LINE(ST(139),SR(139))-(ST(13
2),SR(132)),PSET
2Ø1 T=146:U=142:V=138:W=136:G=13
9:H=145:GG=132:HH=137:SK=Ø
2Ø2 GOTO52
2Ø3 G=147:H=15Ø
2Ø4 GOTO57
2Ø5 T=165:U=162:V=158:W=155:GG=1
59:HH=164:G=152:H=157:SK=Ø
2Ø6 GOTO52
```

GAME

16K ECB

TAKE THE PLUNGE WITH COCO

by Timmy Jones

While watching the diving competition during the 1984 Olympics, I developed the idea for 10-Meter Platform Diving.

To play the game choose forward or reverse dives, then press the down-arrow key to begin the dive and the space bar to finish the dive. You will have five dives to reach the highest possible score. Each dive is randomly scored (from one to 60) based on the difficulty of the dive. With a little practice, you will be doing dives even Greg Louganis would envy.

For harder-to-hit, perfect dives, insert a POKE 65495,0 in the begining of the program.

As a bonus, a similar program, Springboard Diving, will be included on this month's RAINBOW ON TAPE.

80	149
290	118
530	239
750	114
1000	21
1210	251
1380	44
END	167

The listing: PFRMDIVE

```
1 '**************************
2 '**  1Ø METER PLATFORM    **
3 '**        DIVING         **
4 '**                       **
5 '** BY TIMMY JONES        **
6 '**      P.O. BOX 7938    **
7 '**      CLINTON, LA 7Ø722**
8 '**************************
1Ø CLEAR 1ØØ
2Ø CLS(7):PRINT @ 132,"1Ø METER
PLATFORM DIVING";:PRINT @ 172,"D
IVING";:PRINT @ 36Ø,"BY TIMMY JO
NES";
3Ø PMODE 3,1:PCLS:SCREENØ,Ø
4Ø FOR LL=1 TO 6Ø:X=RND(56)+1ØØ:
Y=RND(1Ø)+14Ø:PSET(X,Y,3):NEXT L
L
5Ø SP$="BM 1ØØ,15ØC3E2F2E2F2E2F2
E2F2E2F2E2F2E2F2E2F2E2F2E2F2E2F2
E2F2E2F2E2F2E2F2;BU4BL2ØL1BL4L1B
L4L1BL4L1"
6Ø CD$="BM1ØØ,9ØC2L3D3R1C3U2R2D2
L1U1D3L1D2C4L1D5R1U5R1D5R1U5R2C3
R4L5C4D5R1C3R3D2F1D1R2"
7Ø S1$="C3F2D8C4L1D5R1U5R1D5R1U5
L1D4C3D16U16L1D8L1U2D3C2R2C3R1C2
U3"
```

```
80 LF$="BM100,20S4C2L3D3R1C3U2R2
D2H1D3H1D2C4L1R4D1L4D1R4D1L4D1R4
D1L4D1R4L2C3D8F1"
90 T1$="C2R4D3L1C3U2L1D2L1U2L2D1
R2L3C4D1U3L5D1R4G1L3D1R4C3U1L1R1
D3L4D2U3R3"
100 MS$="BM200,47 C2L3D3R1C3U2R2
D2D1L2D2E1D2C4R1D5L3U5R1D6C3D6R3
L2U7C4U5R2C3R4"
110 DRAW"BM14,100;A0;XT1$;"
120 DRAW"BM14,129;A1;XT1$;"
130 DRAW"BM10,30;A3;XT1$;"
140 DRAW"BM10,60;A2;XT1$;"
150 DRAW"BM50,30;A0;XS1$;"
160 DRAW MS$
170 DRAW CD$
180 DRAW SP$
190 DRAW LF$
200 DRAW"BM 50,80;A1;XS1$;"
210 DRAW"BM 50,120;A3;XS1$;"
220 FOR Y=56 TO 60:PSET(5,Y,2):N
EXT
230 FOR Y=26 TO 30:PSET(8,Y,2):N
EXT
240 DIM CD(18),BS(22),SS(18),AA(
18),AB(140),AC(40),S1(30),S2(40)
,S3(30),S4(26),Q(30),T1(18),T2(1
8),T3(20),T4(20),MS(28),Z(28)
250 GET(92,88)-(114,108),CD,G
260 GET (46,14)-(60,60),S1,G
270 GET (100,130)-(140,150),BS,G
280 GET (110,140)-(140,150),SS,G
290 GET (14,64)-(58,86),S2,G
300 GET (46,100)-(86,120),S3,G
310 GET (90,4)-(108,42),S4,G
320 GET (190,40)-(210,66),MS,G
330 GET (190,10)-(250,30),AC,G
340 GET (200,100)-(220,124),Q,G
350 GET (200,10)-(210,46),AA,G
360 GET (190,42)-(210,68),Z,G
370 GET (0,12)-(22,42),T4,G
380 GET (0,38)-(22,68),T3,G
390 GET (0,84)-(22,114),T1,G
400 GET (0,108)-(22,138),T2,G
410 PCLS
420 FOR X=144TO218 STEP36
430 CIRCLE(X,30),20
440 CIRCLE(X,31),21
450 NEXTX
460 FORX=162TO200 STEP36
470 CIRCLE(X,55),20
480 CIRCLE(X,56),21
490 NEXTX
500 CLS:INPUT"INSTRUCTIONS (Y/N)
";LL$
510 IF LL$="Y" THEN 520 ELSE 540
520 CLS:PRINT:PRINT"  INSTRUCTIO
NS:  YOU WILL HAVE  FIVE DIVES T
O SCORE AS HIGH AS  POSSIBLE. EA
CH DIVE IS SCORED ONA SCALE FROM
 1 TO 60. CHOOSE    FORWARD OR R
EVERSE DIVES, THEN  PRESS (DOWN
ARROW) TO BEGIN DIVEPRESS THE SP
ACE BAR TO FINISH   THE DIVE.
530 I$=INKEY$:IF I$=""THEN 530
540 GOTO 1500
550 DRAW"C3":FORX=0TO3:Y=30:LINE
(X,Y)-(X+30,Y+10),PSET,B:Y=Y+1:N
EXT X
560 DRAW"C3":LINE(30,170)-(255,1
91),PSET,B:PAINT(34,174),3,3
570 DRAW"C2":LINE(0,40)-(30,191)
,PSET,B:PAINT(2,172),2,2
580 DRAW"C2":LINE(30,40)-(70,50)
,PSET,B:PAINT(35,45),2,2
590 GET (50,170)-(250,190),AB,G
600 PUT (58,13)-(78,39),MS,PSET
610 SCREEN1,0
620 A$=INKEY$:IF A$<>CHR$(10) TH
EN GOTO 620
630 IF ZX=2 THEN 1270
640 X=72:Y=0:F=2
650 PUT(58,13)-(78,39),Q,PSET
660 FOR P=1 TO 4
670 F=F+F:IF F=>14 THEN F=14
680 Y=Y+F:X=X+1
690 PUT(X,Y)-(X+22,Y+30),T1,PSET
700 FOR O=1TO90:NEXTO
710 A$=INKEY$:IF A$=" "THEN 920
720 Y=Y+F:X=X+1
730 PUT(X,Y)-(X+22,Y+30),T2,PSET
740 FOR O=1TO90:NEXTO
750 A$=INKEY$:IF A$=" "THEN 970
760 Y=Y+F:X=X+1
770 PUT(X,Y)-(X+22,Y+30),T3,PSET
780 IF P=4 THEN 870
790 FOR O=1TO90:NEXTO
800 A$=INKEY$:IF A$=" "THEN 1070
810 Y=Y+F:X=X+1
820 PUT(X,Y)-(X+22,Y+30),T4,PSET
830 FOR O=1TO90:NEXTO
840 A$=INKEY$:IF A$=" "THEN 1020
850 NEXT P
860 REM ******* BAD DIVE *****
870 PUT(X,150)-(X+40,170),BS,PSE
T
880 PUT(X-10,150)-(X+50,170),AC,
PSET
890 PUT(50,170)-(250,190),AB,PSE
T
900 PLAY"T2AD":SD=RND(9)+10:GOTO
 1210
910 REM ********* S1 *********
920 X=X+6:Y=Y+2
930 PUT(X,Y)-(X+14,Y+46),S1,PSET
940 IF Y+42=>170 THEN 1120
950 X=X+0:Y=Y+10:GOTO 930
960 REM******* S2 ********
```

```
 97Ø X=X+4:Y=Y+6
 98Ø PUT(X,Y)-(X+44,Y+32),S2,PSET
 99Ø IF Y+3Ø =>17Ø THEN 86Ø
 1ØØØ X=X+Ø:Y=Y+1Ø:GOTO 98Ø
 1Ø1Ø REM ********* S3 ********
 1Ø2Ø X=X+4:Y=Y+6
 1Ø3Ø PUT(X,Y)-(X+4Ø,Y+2Ø),S3,PSE
T
 1Ø4Ø IF Y+3Ø=>17Ø THEN 86Ø
 1Ø5Ø X=X+Ø:Y=Y+1Ø:GOTO 1Ø3Ø
 1Ø6Ø REM ******* S4 ********
 1Ø7Ø X=X+4:Y=Y+2
 1Ø8Ø PUT(X,Y)-(X+18,Y+38),S4,PSE
T
 1Ø9Ø IF Y+3Ø=>17Ø THEN 87Ø
 11ØØ X=X+Ø:Y=Y+1Ø:GOTO 1Ø8Ø
 111Ø REM ****** GOOD DIVE *****
 112Ø REM
 113Ø PUT(X,Y)-(X+22,Y+22),AA,PSE
T:PUT(X-1Ø,16Ø)-(X+2Ø,17Ø),SS,PS
ET
 114Ø PUT(X-1Ø,15Ø)-(X+2Ø,17Ø),AC
,PSET
 115Ø PUT(5Ø,17Ø)-(25Ø,19Ø),AB,PS
ET
 116Ø PLAY"T9EFDDAGG"
 117Ø IF P=1 THEN SD=RND(9)+1Ø:GO
TO 121Ø
 118Ø IF P=2 THEN SD=RND(9)+2Ø:GO
TO 121Ø
 119Ø IF P=3 THEN SD=RND(9)+3Ø:GO
TO 121Ø
 12ØØ IF P=4 THEN SD=RND(2Ø)+4Ø:G
OTO 121Ø
     TOTAL SCORE IS :";SC:PRINT:
INPUT"DO YOU WANT TO PLAY AGAIN(
Y/N)";Q$:IF Q$<>"Y"THEN END ELSE
 RUN
 125Ø FOR G=1 TO 1ØØØ:NEXT G
 126Ø GOTO15ØØ
 127Ø Y=1Ø:X=72:F=2:PUT(58,13)-(7
8,39),Q,PSET:PUT(X,Y)-(X+22,Y+22
),CD,PSET
 121Ø FL=FL+1:PRINT"  DIVE #:";FL
 122Ø PRINT"          SCORE LAST D
IVE:";SD
 123Ø SC=SC+SD
 124Ø IF FL=5 THEN PRINT"
 ================ ==":PRINT"

 128Ø Y=Y-2
 129Ø FOR P=1 TO 4
 13ØØ F=F+F:IF F=>14 THEN F=14
 131Ø Y=Y+F:X=X+1
 132Ø PUT(X,Y)-(X+22,Y+32),T3,PSE
T
 133Ø FOR O=1TO9Ø:NEXT O
 134Ø A$=INKEY$:IFA$=" "THEN X=X-
2:GOTO 92Ø
 135Ø Y=Y+F:X=X+1
```

```
136Ø PUT(X,Y)-(X+22,Y+32),T2,PSE
T
137Ø FOR O=1TO9Ø:NEXTO
138Ø A$=INKEY$:IF A$=" "THEN 1Ø2
Ø
139Ø Y=Y+F:X=X+1
14ØØ PUT(X,Y)-(X+22,Y+32),T1,PSE
T
141Ø IF P=4 THEN 87Ø
142Ø FOR O=1TO9Ø:NEXTO
143Ø A$=INKEY$:IF A$=" "THEN 1Ø7
Ø
144Ø Y=Y+F:X=X+1
145Ø PUT(X,Y)-(X+22,Y+32),T4,PSE
T
146Ø FOR O=1TO9Ø:NEXTO
147Ø A$=INKEY$:IF A$=" "THEN 97Ø
148Ø NEXT P
149Ø GOTO 87Ø
15ØØ CLS:PRINT"             TYPE OF
 DIVE":PRINT:PRINT"          (1) F
ORWARD":PRINT"         (2) REVERS
E"
 151Ø PRINT"          (1-2)";
152Ø A$=INKEY$:IF A$<"1" OR A$>"
2" THEN 152Ø
153Ø IF A$="1" THEN ZX=1 ELSE IF
  A$="2" THEN ZX=2
 154Ø CLS:GOTO 55Ø
```

WISHING WELL   16K ECB   32K ECB

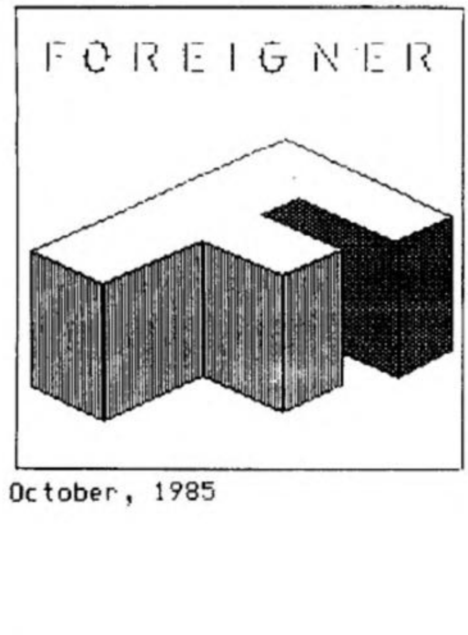
Rock'n'Rollin' Back To The CoCo Rockfest

By Fred B. Scerbo

During the past year I have received a great deal of mail asking when readers could expect the next installment of two of my more popular graphics programs, *CoCo Rockfest* and *Baseball Fever*. After over a year of waiting, *Rockfest II* is now a reality, and what better time for its release than as part of THE RAINBOW's anniversary issue?

In this version, I have included some of the symbols and designs for the most requested groups and have also included some of the graphics breakthroughs I have introduced to you in these pages over the last few months.

Before we get into the details of how *Rockfest II* works, let me deal with the issue of *Baseball Fever II*. I have received many letters from individuals, some of whom get quite angry that I have not replied to say when they could expect to see *Fever II*. Let me state once again: I simply do not have the time to answer all this mail, create the programs and write the column. If you want to know what is coming in following issues, just read the column. (I know I still owe a few people the 32 bracket wrestling programs they requested with a SASE. I will forward those after I add a few more refinements, so please be patient.)



The Wish

For those who may not have seen our first *Rockfest*, let me explain what the program is all about. As I have stated many times before, the Extended Color BASIC Manual which comes with your CoCo is one of the most powerful and flexible graphics languages available. In fact, recently, a friend who is a strong supporter of the Commodore 64 came to me to show me the specifications on the new Commodore 128K computer. This new C128 includes graphics commands such as DRAW, PAINT, LINE and COLOR. Naturally, my response was, "Big deal! CoCo users have had those commands for over four years now!" I guess our little Color Computer has been way ahead of its time!

When I released the first *Rockfest*, I stated that young programmers would find it very interesting to be able to type in graphics commands from the magazine listing and see the symbols and logos from their favorite rock groups explode in color on their computer screens. This helps encourage youngsters to explore how these commands work while improving their keyboard skills as well.

Needless to say, *Rockfest I* was a big success. I have even been approached by owners of discos and night clubs about using the CoCo to project these images on their video screens. (However, no one has actually done it yet, to my knowledge.) The mail I have received regarding the sequel has been overwhelmingly in favor of drawing the logos for some of the more popular "heavy metal" bands. With that in mind, here are your wishes granted in *Rockfest II*.

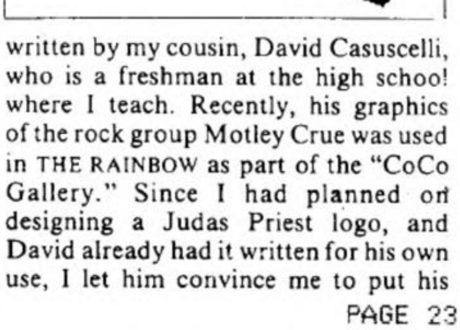
The Program

The remainder of this article will be much smaller than usual since *Rockfest II* really does not break any new ground as far as graphics techniques are concerned. The one major difference between this version and the original is this version will fit into a 16K machine simply by leaving out the curtain and stage title card, and the remark lines.

This will be a big plus for all 16K owners, since the last version would only allow you to type in portions of the program, one group at a time. *Rockfest II* will run perfectly in 32K/64K without any modifications. I will explain the modifications for 16K at the end of this article.

The eight groups included in this version are Judas Priest, Foreigner, Iron Maiden, Ratt, Def Leppard, Big Country, Kiss and The Police. (Not all are heavy metal!) Some of these logos will only take a few lines, such as Ratt, which only takes six lines, or Kiss, which only takes 10 lines. Sure, there is no real work of art, such as the ZZTop logo from *Rockfest I*, and the logo for The Police is a rather simple design from the *Synchronicity* album, but what do you expect? ZZTop did take a great deal of time!

For the first time I have also included a graphics which I did not write the lines for. The Judas Priest logo was written by my cousin, David Casuscelli, who is a freshman at the high schoo! where I teach. Recently, his graphics of the rock group Motley Crue was used in THE RAINBOW as part of the "CoCo Gallery." Since I had planned on designing a Judas Priest logo, and David already had it written for his own use, I let him convince me to put his



version in *Rockfest II.* (It didn't take too much convincing. As I've said before, no use reinventing the wheel.) I think you will find David did quite a good job on it.

Another feature which I included in this version is the use of PCOPY to help create the actual graphics. You will notice that after the Ratt logo is drawn at the top of the screen, it is repeated three more times rapidly to fill the screen with four copies of the logo. This technique is also used on the Police logo to repeat the graphics for the paint brush stroke. Not only does this save us some program lines, it helps to draw the graphics on the screen much more rapidly.

Those who are familiar with my graphics in *Football Fever I* and *II* and the extra PMODE4 colors will notice I have used one color from those techniques: yellow. I have put this into one array and use it to paint yellow using the OR command with the PUT statement. For those who wish to know how these colors are created, please refer to those articles (November 1984, Page 177 and December 1984, Page 107).

Some of you will remember that I also used the PCOPY command with the POKE178,n method to rapidly page through different-colored versions of AC-DC and Asia in the last version.

This time I used the same technique for Kiss, only I created the graphics in PMODE1 rather than PMODE3 or 4. This allows us to use the PCOPY command without having to PCLEAR any additional pages.

Running the Program

After you have successfully typed in the listing, RUN the program. The screen will fill with a curtain and stage, draw out the title "CoCo Rockfest II" and then draw a set of A's and B's at the bottom of the screen.

Of the two letters drawn, press the one which is colored red and the program will proceed to the menu. Press the letter next to the graphics you wish to see, and the screen will proceed to draw the group which you selected. If you wish to return to the menu, simply press the ENTER key and the menu will reappear.

If you wish to have a little fun with the program, type AUDIOON:MOTORON and ENTER before running the program. You may now place a music cassette in your computer cassette recorder and play the music over the speaker of your TV.

Also, you may make screen dumps of these graphics if you have a screen dump program. (Please don't write and ask me to send you mine. I don't have one. I'm using someone else's!) Select the graphics you want from the menu, press BREAK and then load your graphics dump.

16K Modifications

If you wish to fit this program into 16K, then delete the following lines from this program listing:

```
DEL 24 - 82
DEL 112 - 128
DEL 248 - 256
DEL 324 - 336
DEL 444 - 454
DEL 514 - 528
DEL 638 - 652
DEL 762 - 776
DEL 826 - 840
```

If you have RAINBOW ON TAPE and only have 16K, you may still load the program, but first you must PCLEAR1. Next, load the program, delete the lines above, then PCLEAR4 and RUN. You may also wish to insert the following lines:

```
24 PMODE4,1:PCLS1:SCREEN1,1:
   PMODE3:PCLS2
26 X$=INKEY$:IF X$<>CHR$(13)
   THEN 26
```

On running the program, the screen should be red. If it is not, press the Reset button and RUN again until it is red. If the screen is red, simply press the ENTER key and the menu will appear. You will not get the title card, but this is better than nothing.

50	153	508	238
90	210	618	41
210	166	654	7
230	38	730	200
306	77	762	248
338	128	824	207
430	46	END	26

The listing: RCKFEST2

```
10 '****************************
12 '*   THE COCO ROCK-FEST #2   *
14 '*     BY FRED B. SCERBO     *
16 '*     COPYRIGHT (C) 1985    *
18 '*149 BARBOUR ST.N.ADAMS,MA.*
20 '****************************
22 DIM A(20)
24 PMODE4,1:PCLS0:SCREEN1,1
26 LINE(0,170)-(255,174),PSET,B
28 LINE(0,174)-(255,192),PSET,BF
30 PMODE3,1
32 POKE178,55:PAINT(128,2),,4
34 FORI=1TO40STEP4:IK=RND(3)+1:C
OLOR K,K::LINE(I,0)-(I,170),PSET
:LINE(254-I,0)-(254-I,170),PSET:
NEXT
36 FORI=2TO120STEP2:K=RND(3)+1:C
IRCLE(0,0),I,K,.4,0,.25:CIRCLE(2
55,0),I,K,.4,.25,.5:NEXT
38 LINE(70,48)-(182,120),PSET,BF
40 DRAW"BM108,52C3R4ND6R4BR6D6U3
R6D3U6BR6NR6D3NR6D3NR6"
42 PMODE4,1
44 FORI=96TO140STEP44
46 FOR Y=6TO10
48 CIRCLE(I,70),Y,0,.9,.15,.9
50 CIRCLE(I+11,78),Y,0,.9,.6,.8
52 NEXT Y
54 FOR Y=6TO10
56 IF I=96 THEN 58 ELSE 60
58 CIRCLE(I+33,78),Y,0,.9,.6,.8
60 CIRCLE(I+20,70),Y,0,.9
62 NEXTY
```

```
64 NEXTI
66 PMODE3,1
68 DRAW"BM76,94C2U12R4F2D2G2L4R4
F2D4BR8H2U8E2R4F2D8G2L4BR14R4NE2
L4H2U8E2R4NF2BR8D12U6R2E4NU2G4F4
D2BR4BU6R6BD6BR6U6NR4U6R6BR6"
70 DRAW"NR6D6NR4D6R6BR6R8U6L8U6R
8BR4R4ND12R4BD20BL46C3L20R4D12L4
R20L4U12"
72 FORI=6TO120STEP16:I$=STR$(I):
Y$=STR$(256-I)
74 DRAW"BM"+I$+",188C3U4NR4U4R4D
8"
76 DRAW"BM"+Y$+",188C2NL6U4NL4U4
L6R2D8"
78 NEXTI
80 X$=INKEY$:IFX$="A"THEN82ELSEI
FX$="B"THEN84ELSE80
82 B=2:R=3:R$="C3":GOTO86
84 B=3:R=2:R$="C2"
86 CLS0:PMODE4,1:PCLS1:SCREEN0,0
:PMODE3:DRAW"S4BM0,1":FORI=1TO64
:DRAWR$+"RBRC4R2":NEXTI:DRAW"BM2
,2":FORI=1TO64:DRAWR$+"RBRC4R2":
NEXTI:GET(0,1)-(256,2),A,G
88 CLS5:SCREEN0,0:PRINT@132," A)
 JUDAS PRIEST        ";
90 PRINT@164," B) FOREIGNER
     ";
92 PRINT@196," C) IRON MAIDEN
     ";
94 PRINT@228," D) RATT
     ";
96 PRINT@260," E) DEF LEPPARD
     ";
98 PRINT@292," F) BIG COUNTRY
     ";
100 PRINT@324," G) KISS
     ";
102 PRINT@356," H) POLICE
     ";
104 X$=INKEY$:IFX$=""THEN104
106 X=ASC(X$)-64:IF X<1 THEN 104
 ELSE IF X>8 THEN104
108 ON X GOSUB200,300,400,500,60
0,700,800,900
110 GOTO88
112 '
114 '
116 '
118 '
120 '
122 '
124 '
126 '
128 '
200 'JUDAS PRIEST
202 'BY DAVE CASUSCELLI
204 PMODE4,1:PCLS1:SCREEN1,1:PMO
DE3
206 DRAW"BM0,0C0"
208 DRAW"BM21,48S4C1NM+0,+37BD1M
+26,-21'M+00,+55'M-11,-4'M+08,-8
U6'M-13,+13D4'M+21,+8'U57U3M-5,-
1D3'BM-17,+46'M-28,+22D4M+28,-22
M-28,+22M+22,+8M+21,-15BU3
210 DRAW"BM+8,-50M+5,-3D19M-5,+3
NBF4M+5,-3NF2U19M+6,+2D15M+8,+8M
+0,-20NM-7,+3NM+5,+1BM+6,+2M+0,+
17M+6,+6M+0,-21NM-6,+4NM+9,-4M+5
,+1M+5,+4M+0,-9M+5,+1M+0,+29F2E4
U17GLGLGEREREBM+0,+48BM-15,-32FE
3U11HLHLD15BU2
212 DRAW"BM+16,-14M+4,+1M+4,+4RU
3M+6,+2D19BL10NU16FE3U12HLH"
214 DRAW"BR9BD14D5M+14,+4M-8,-18
NM-5,+2U5NM-5,+2M+15,+5D5NM-4,+4
M-08,-3M+9,+20D3NG3BM-13,-4"
216 A$="M+11,+11M+5,-2M+6,+6M+3,
-3M+7,+7M+3,-3M+7,7M+4,-3M+6,+6M
+3,-3M+7,7M+3,-3M+14,+4"
218 DRAW"BM-70,-28"+A$+"M+3,+1
220 DRAW"BM150,58NM-14,+9D40NG3F
4NG4E2U34M+13,+5M-11,+9D6M+16,-1
3U5M150,58"
222 DRAW"BM+20,+21NM-14,+10D5M+6
,-3D6M-6,+3D14G2NG3H4NG3U20BM+15
,-13M+5,+2M-0,+5NM-7,+6M-5,-2NM-
10,+8U5LGLG"
224 DRAW"BM+5,+10NG3ND25M+6,+2D2
1M+6,+2U20NM-6,+3M+18,+6D5M-11,+
11BM-1,-8U7M+6,+2M-6,+6BM+1,+7M+
23,+7M-9,-17NM-10,+10U5NG2M+18,+
6D5NG5M-09,-3LM+9,+17D4NG3M-44,-
13GGGGHUHUNG3U2"
226 G$="M+11,+11M+5,-2M+6,+6M+3,
-3M+7,+7M+3,-3M+7,7M+4,-3M+6,+6M
+3,-3M+7,7M+4,-4M+19,+5M+9,+9M+5
,-5M+10,+10M+5,-5M+11,+11M+5,-5M
+47,+14"
228 DRAW"BM-122,-39NM-5,+3D6NM-5
,+3U6"+G$:DRAW"BM-178,-54M+11,+1
1M+5,-2M+6,+6M+3,-3M+7,+7M+3,-3M
+7,+7M+4,-3M+6,+6M+3,-3M+8,+8M+6
,-6M+14,+4M+12,+12M+5,-5M+10,+10
M+5,-5M+11,+11M+6,-6M+55,+16"
230 DRAW"U28M+9,+3NG9U5M-9,-3U10
M-5,-1D9M-5,-1NG2D5NG6M+5,+1NG9D
23M-3,-1"
232 DC$="M+3,+3M+3,-3M+8,+8M+4,-
4M+9,+9M+4,-4M+9,+9M+4,-4M+8,+8M
+4,-4M+8,+8M+5,-5M+10,+10M+5,-5M
+10,+10M+6,-6M+10,+10M+5,-5M+10,
+10M+5,-5M+48,+14M+20,-16":DRAW"
BM-191,-30"+DC$
234 PMODE4
236 PAINT(120,112),0,0:PAINT(74,
74),0,0:PAINT(186,112),0,0:PAINT
(150,101),0,0:PAINT(165,105),0,0
:PAINT(199,97),0,0:PAINT(227,105
),0,0:PAINT(229,111),0,0:PAINT(2
49,113),0,0:PAINT(139,75),0,0
238 PMODE4:SCREEN1,1:PMODE3
240 PAINT(49,64),R,1:PAINT(60,44
),R,1:PAINT(152,75),R,1:PAINT(18
1,71),R,1:PAINT(174,86),R,1:PAIN
T(183,88),R,1:PAINT(246,107),R,1
242 PAINT(0,0),B,1
244 PAINT(200,95),1,1:PAINT(160,
74),1,1:PAINT(180,77),1,1:PAINT(
234,106),1,1
246 X$=INKEY$:IFX$=CHR$(13)THENR
ETURNELSE246
248 '
250 '
252 '
254 '
256 '
258 'FOREIGNER
300 PMODE4,1:PCLS1:SCREEN1,1:PMO
DE3
302 DRAW"C1S4":GOSUB304:GOTO310
304 DRAW"BM140,54M-159,46D60M+45
,+15M+60,-20M-60,+20U60M-45,-15M
+45,+15M+60,-20ND60M+48,+16D60M-
48,-16M+48,+16"
306 DRAW"M+36,-12U60M-36,+12M+36
,-12M-48,-16M+21,-7M+60,+20D60M-
33,-11M+33,+11M+36,-12U60M-36,+1
2M+36,-12M-100,-34M-12,+4
308 RETURN
310 PAINT(190,100),1,1:PAINT(240
,100),1,1
312 FORI=50TO180STEP2:PUT(0,I)-(
256,I+1),A,OR:NEXT:GOSUB304
314 PAINT(4,160),B,1:PAINT(80,11
0),R,1:PAINT(130,110),R,1:PAINT(
160,110),R,1:PAINT(140,56),1,1
316 PAINT(1,1),1,1:PAINT(126,188
),1,1:DRAW"C4":GOSUB304
318 DRAW"S4C4BM10,20U12BR4NR6BD6
BR2R2":CIRCLE(40,14),8,4:DRAW"BU
8C1R32D16NL28NH28R2NH28BR12BU2C4
U12BR4R6F2D2G2L2M+4,+6BR18U12BR4
NR6BD6BR2R2BD6NL4R2BR18U12"
320 CIRCLE(146,14),8,4,1,1,.85:D
RAW"C1BU2R32D16NL28BH8NH6C4R8BR1
8ND6U2BU4F12BU6U6BD12BR18U12BR4N
R6BD6BR2R2BD6NL4R2BR16C4U12BR4R6
F2D2G2L2M+4,+6"
322 X$=INKEY$:IFX$=CHR$(13)THENR
ETURNELSE322
324 '
326 '
328 '
330 '
332 '
334 '
336 '
338 'IRON MAIDEN
```

```
400 PMODE4,1:PCLS0:SCREEN1,1:PMO
DE3,1
402 DRAW"S3C4BM46,70U56R16D56NL1
6"
404 DRAW"BR8U56R14D10E12F28G14F2
8G10H44BU8E10F10L20BD8D24L14"
406 DRAW"BR72H18E40F40BD3BL20L40
E20F20BR20BU3G18L48"
408 DRAW"BR72U46H12R20F24U24R16D
78H16U16H16D28L16"
410 DRAW"BM12,140U46H12R26F16E16
F16D64H16U40G16H14D32L16"
412 DRAW"BR80H12E10H8E14F8E22R16
D56L26H10G10BE22BU2E8D16H8BD2BG2
2"
414 DRAW"BR54U56R16D56NL16"
416 DRAW"BR8U56R16F38G19L36BE15B
R2R18H18D18BL2BG15
418 DRAW"BR72H12U44R34D12L18D10R
18D12L18D10R18D14L18"
420 DRAW"BL36BR72U44H12R20F24U24
R16D76H16U16H16D28L16"
422 PAINT(50,30),R,4
424 PAINT(70,30),R,4
426 PAINT(132,30),R,4
428 PAINT(170,30),R,4
430 PAINT(10,100),R,4
432 PAINT(100,100),R,4
434 PAINT(120,100),R,4
436 PAINT(140,100),R,4
438 PAINT(180,100),R,4
440 PAINT(212,100),R,4
442 X$=INKEY$:IFX$=CHR$(13)THENR
ETURNELSE442
444 '
446 '
448 '
450 '
452 '
454 '
456 'RATT
500 PMODE4,1:PCLS1:SCREEN1,1:PMO
DE3
502 DRAW R$:DRAW"S4BM20,16R16D30
L16U30BU6H12R46"
504 CIRCLE(56,10),20,R,.7,.77,.2
0:DRAW"BD22BR8F14R4F10L24H20U6H6
R14H6L16F4L14H4BR66BD22M+16,-32R
26M+23,+46L50H12R30F4R4M-12,-24L
4M-8,+16L16"
506 DRAW"BR50BU32R50D46L18U28E2U
2H2L2G2D4L18M-9,-18BR58R50M-8,+1
8L20U4H2L2G2D2F2D28L18U46"
508 PAINT(22,18),R,R:PAINT(22,2)
,R,R:PAINT(120,18),R,R:PAINT(180
,18),R,R:PAINT(238,2),R,R
510 PCOPY1TO2:PCOPY1TO3:PCOPY1TO
4
512 IFINKEY$<>CHR$(13)THEN512ELS
ERETURN
514 '
516 '
518 '
520 '
522 '
524 '
526 '
528 '
600 'DEF LEPPARD
602 PMODE4,1:PCLS1:SCREEN1,1
604 DRAW"C0":GOSUB606:GOTO628
606 DRAW"S5BM90,100U80M+20,+40M-
20,+40M+20,-40BL8M-6,-14D26M+6,-
14BR11BD2M+14,-34D16M-6,+15R6M+3
,+5L8M+6,+15D16M-14,-34"
608 DRAW"BM+1,+1BD3BR19NR4M-3,-5
R7U37M+14,+28L6M-3,-5D13R11M+4,+
6L15D36M-6,-11U26"
610 DRAW"BR22BM16,146BD18NR32M+3
0,-60D17M-19,+38R18M+3,+6BU21"
612 DRAW"BL4M+14,-34D16M-6,+15R6
M+3,+5L8M+6,+15D16M-14,-34BR19BD
23"
614 FOR I=1TO2
616 DRAW"NU60R7U20BU6U14M+7,+14L
7BD6R18M-24,-40BD60BR27"
618 NEXT I
620 DRAW"BL8M+27,-54D54L6U17BU4U
7M-4,+7R4BD4L6M-8,+17L8BR32"
622 DRAW"NU60R7U20BU6U14M+7,+14L
7BD6R4M+10,+20R7M-10,-20R8M-24,-
40BR32BD90"
624 DRAW"U80M+20,+40M-20,+40M+20
,-40BL8M-6,-14D26M+6,-14"
626 RETURN
628 PAINT(94,10),0,0:PAINT(134,1
8),0,0:PAINT(150,12),0,0
630 PAINT(52,100),0,0:PAINT(66,1
10),0,0:PAINT(74,106),0,0:PAINT(
110,106),0,0:PAINT(160,110),0,0:
PAINT(170,100),0,0:PAINT(210,100
),0,0
632 FORI=1TO191STEP2:PUT(0,I)-(2
56,I+1),A,OR:NEXT
634 PMODE3:DRAWR$:GOSUB606
636 IFINKEY$<>CHR$(13)THEN636ELS
ERETURN
638 '
640 '
642 '
644 '
646 '
648 '
650 '
652 '
654 'BIG COUNTRY
700 PMODE4,1:PCLS1:SCREEN1,1
702 DRAW"S4C0BM16,110R60E6U6H6E6
U6H6L60D6R8D24L8D6BR30BU6U10NR10
BU4U10R10D10NL10BD4D10L10"
704 DRAW"C0BM128,110R30U6L12U24R
12U6L60D6R12D24L12D6R30"
706 DRAW"BM238,110L12H4G4L22"
708 CIRCLE(206,90),36,0,.6,.3,.6
5
710 DRAW"C0BM184,74R30F4U4R14D14
L18H4L8"
712 CIRCLE(212,92),19,0,.5,.25,.
67
714 DRAW"BM213,101E3L6U6R34D6L6D
12"
716 DRAW"BM20,120NR8G4D40F4R12E4
U16L8D16L4U40R4D16R8U20L4G2H2"
718 DRAW"BR16NR8G4D40F4R12E4U40H
4L4BD4D40L4U40R4BU4"
720 DRAW"BR12NR8D4R4D40F4R14E4U4
0R4U4L12D4R4D38G2L8U40R3U4L6"
722 DRAW"BM108,120L10D4R4D40L4D4
R12U4L4U30M116,164F4R6E4U40R4U4L
12D4R4D30M114,120L6"
724 DRAW"BR31NR34D18R9U14R4D40L4
D4R16U4L4U40R4D14R9U18"
726 DRAW"BR5NR20D4R4D40L4D4R18U4
L4U18R6D18F4R10U4L4U16H4E4U14H6L
4BD4L6D16R6U16BU4"
728 DRAW"BR16R12D4L4M229,140M236
,124L4U4R12D4L2M234,146D18R4D4L1
7U4R4U18M216,124L2U4"
730 FORI=46TO0STEP-6:LINE(128,46
)-(0,I),PRESET:NEXT
732 FORI=3TO255STEP10:LINE(128,4
6)-(I,0),PRESET:NEXT
734 FORI=4TO46STEP6:LINE(128,46)
-(255,I),PRESET:NEXT
736 CIRCLE(128,44),26,0,.8
738 PAINT(128,48),0,0
740 PMODE3,1
742 CIRCLE(128,44),23,4,.8
744 PAINT(128,44),4,4
746 PMODE4,1
748 FORI=23TO25:CIRCLE(128,44),I
,0,.8:NEXT
750 PAINT(2,56),0,0
752 PAINT(50,81),0,0:PAINT(50,10
2),0,0
754 PAINT(50,125),0,0:PAINT(195,
125),0,0
756 FORI=8 TO255STEP2:LINE(I,0)-
(I,192),PRESET:NEXT
758 FORI=0TO192STEP2:LINE(0,I)-(
255,I),PRESET:NEXT
760 IFINKEY$<>CHR$(13)THEN760ELS
ERETURN
762 '
764 '
766 '
768 '
770 '
772 '
774 '
```

```
776 '
778 'KISS
800 CLS0
802 PMODE1,1:PCLS1:SCREEN0,0:PMO
DE1,3:PCLS1:SCREEN1,1
804 FORI=5TO255
806 PMODE1,1:PCLS1
808 DRAW"S4C2BM0,136U76R28ND30":
CIRCLE(26,62),26,2,1.2,.03,.23:D
RAW"BR22ND6NR28":CIRCLE(42,58),4
0,2,1.2,.03,.16
810 DRAW"BD38BR10NE2NF2BD38":CIR
CLE(42,138),40,2,1.2,.83,.99:DRA
W"NR20L10":CIRCLE(26,138),26,2,1
.2,.79,.99:DRAW"BL22NU26L28BR82U
76R26D76NL26"
812 FORG=1TO2:DRAW"BU36E40R36G30
R34G46L34E36L28NL6BD36BR62BU36E4
0R36G30R34G46L34E36L30"
814 POKE178,1
816 PAINT(10,62),,2:PAINT(100,62
),,2:PAINT(160,62),,2:PAINT(240,
62),,2
818 PCOPY1TO3:PCOPY2TO4
820 PMODE1,3:SCREEN1,(RND(2)-1)
822 IFINKEY$<>CHR$(13)THEN824ELS
ERETURN
824 NEXTI
826 '
828 '
830 '
832 '
834 '
836 '
838 '
840 '
842 'POLICE
900 PMODE4,1:PCLS1:SCREEN1,1:PMO
DE3
902 DRAW"BM34,40C1U20NL8R8BR8D20
U10R14NU10D10BR10NR10U10NR8U10R1
0BR20R12D10L12NU10D10BR20U20R12D
20NL12BR12NU20R12BR10NU20BR10NR1
2U20R12BR10NR12D10NR10D10R12"
904 PMODE4
906 DRAW"BM0,52C0D2F4R4D2L4G2D2F
2D2F4L4G2D2F2G2D2F2D2F2R6D2M+40,
+4R30UR20UR16M+40,+2M+86,-5E2U2H
4U2R2E2U2H2L2H2U2E4R4E2U2H2U2E2U
2H2L14DL16M-90,+4M-40,-2M-20,+3M
-50,-3H2L4H2L4G2L4D2"
908 PAINT(40,58),0,0
910 PCOPY2TO3:PCOPY2TO4
912 PMODE3
914 PAINT(40,58),B,4
916 PAINT(40,106),R,4
918 FORI=144TO192STEP2:PUT(0,I)-
(256,I+1),A,OR:NEXTI
920 IFINKEY$<>CHR$(13)THEN920ELS
ERETURN
```

TAKING BASIC TRAINING

16K ECB

The Goolian Olympics Of Trig Functions

By Joseph Kolar

Creating action plots of trigonometrical functions is not particularly difficult. When displayed on the monitor, the Hi-Res graphics plots are nice to watch as they run their course.

A new concept is introduced wherein a story line is built around each graphics program routine to give it a *raison d' etre*. It is envisioned that an adult supervisor is introducing the wonders of the Color Computer to a small child. The adult is interacting with the child. Here is the procedure.

The adult reads the text to the child, introducing the story plot. He CLOADs the first listing and continues with the story. In the meantime, the child is observing how the adult LOADs a program from cassette. At cue marks in the story, he RUNs the program and continues on with the suggested story. When the viewer's attention span wavers, the adult hits the BREAK key, types NEW and CLOADs the next listing.

The adult continues reading until the cue and runs the program at that point. Since the graphics are stylized, the explanations before viewing are quite important. The child must understand the significance of what he/she is viewing.

Now that the supervising adult knows the procedure, type in CLOAD"ANTHEM" and begin the story.

There is a planet named Gool, far, far away. It is not a very big planet. You can't even see it when you look up at the stars in the night sky. The inhabitants of Gool are called Goolians. Goolians are not like us. They do not speak like us. They look odd, but Gool is an odd place. It is very unlike what Earth people consider normal.

One day, a clever Goolian scanned the heavens with his radio-telescope. He happened to beam in on the 1984 Summer Olympic games in Los Angeles. Goolians have such keen eyesight that they could actually look into the coliseum and see the sports events in progress. The clever Goolian was bemused by the sports events.

Of course, since Goolians have no sports, they do not understand about winning or losing. The clever Goolian liked what he saw. Whatever they were doing on Earth, it sure looked like fun. But, Goolians like to play! The clever Goolian wrote in his notebook about all of the strange things he was astonished to observe.

Unfortunately, it stoned on Gool that day and the notes got all stone marked. You see, it doesn't rain or snow on Gool, it rocks and stones instead. Rock is not too bad because it melts to stone. Stone dents notebooks and makes notes impossible to read.

Even though the clever Goolian failed to beam in on Earth again, he was truly impressed with sports. His stone marked notebook was useless. He wanted to have some Olympic fun, so he and all the other Goolians got together and made up their own games — games that nobody ever wins or ever loses.

(Now RUN and ENTER the program.)

Look at the friendly Goolians. They usually, but not always, walk on stilts. Girl Goolians wear blue stilts. Boy Goolians wear red stilts. Stilts are Goolians' shoes.

The entire Goolian Olympic team is standing in a Goolian straight line. They are singing the "Goolian Planetary Anthem."

Next, hit the BREAK key, type NEW and CLOAD"TRACK".

The first competition is track. Their tracks are quite different from ours. Two runners race around their fourth dimension tracks. Sometimes one track is on top, sometimes the other track in on top and sometimes the two tracks are linked together.

(Now RUN and ENTER.)

Watch the happy Goolians for at least five minutes and you can see their fourth dimension track change colors. They wear special track stilts that blink from blue to red and red to blue. Their word for this type of track is "mobius".

Hit the BREAK key, type NEW and CLOAD"LADDER".

Three fun-loving Goolians compete

in the 1000 meter ladder event.

(RUN and ENTER.)

The Goolians race up and down the fourth dimension ladder. The ladder steps change shape and disappear. Every once in a while, Goolian runners tire near the top of the ladder. A new set of Goolians race up to rescue them and continue the race.

Hit the BREAK key, type NEW and CLOAD"TUGOFWAR".

In the next event, Goolians have a tug-of-war with a giant Goolian musical instrument, the sqwakboks. It is somewhat like an Earth accordian. They get between parts of the sqwakboks and push this way, then push that way. As usual, there are no winners or losers, just tired Goolians.

(RUN and ENTER.)

The tune you hear is a Goolian rock and roll classic. Watch the sqwakboks slowly change color and shape. The song may sound odd to Earth people, but it is number one on the Goolian hit list.

Hit the BREAK key, type NEW and CLOAD"PARAJUMP".

The big event in the Goolian Olympics is the parachute jump. The Goolian sky has a roof. The ground is very soft, as you shall see.

(RUN and ENTER.)

Three Goolians have their chutes hooked onto the Goolian sky with special skyhooks. They are very careful and follow the safety precautions. The rule is to hook up and drop safely! They believe a good Goolian is a live Goolian.

Now, hit the BREAK key, type NEW and CLOAD"PARALIFT".

Have you ever wondered how the happy Goolians got up to the roof of the sky to clip on their parachute skyhooks? RUN and ENTER to find out.

Now you know! The Goolians have rocketpacks on their backs. They have two different ways to reach the sky roof. The Goolian in the middle goes up in quantum leaps — a very special energy. The other two just blast off and rocket up, leaving a polluting dust trail behind them.

Hit the BREAK key, type NEW and CLOAD"BUCKLEUP".

Sometimes there is no galactic wind. Parachuting can become a very dangerous sport, indeed. The chutists drop straight down.

(RUN and ENTER.)

Watch what happens to the bad Goolian who doesn't follow the safety rules. The Goolian in the middle drops with a thud and vaporizes. This is not a nice way to end the Goolian Olympics. Let this be a lesson to all Goolians: buckle up and live!

Note: If you want to save the middle Goolian and give him one more chance, take the REM marker out of Line 225. Here is how to do it: hit BREAK, EDIT 225, press the 'D' key, ENTER and RUN. To replace the REM marker: hit BREAK, EDIT 225, press the 'I' key, press SHIFT and '7' simultaneously and ENTER.

If you want to make a good Goolian out of the bad one: hit the BREAK key, EDIT 216, press the 'I' key, press SHIFT and '7' simultaneously, ENTER and RUN. To unhook him again: hit BREAK, EDIT 216, press the 'D' key and ENTER.

There is no reason why you can't personalize the story line and include the child. You may have your own interpretation of the story; use it, by all means. You may also speak extemporaneously and weave a plot of your own.

A final note: A few of the routines take about 10 minutes to recycle. There is no need to watch the entire routine. Just press BREAK when the child begins to get bored. Also, encourage the child to participate in the operation of the computer. Allow the child to type RUN and hit ENTER. He/she will be thrilled to become manually involved.

Listing 1: ANTHEM

```
0 'GOOLIAN OLYMPICS---PLANETARY
ANTHEM
10 'CREATED BY J.KOLAR, 1984
30 PMODE3:PCLS:PMODE4
40 A=126:B=90:R=72:PI=1.70
50 DIM S(13)
60 DRAW"BM0,0D24R2U3BU2U11F3R3D5
L3G3BD2E3R10F3D3R2U24L2D6BD2D11H
3L3U5R3E3BU2G3L3U3L4D3L3H3U5L2"
61 PAINT(10,7),1,1
70 GET(0,0)-(20,24),S,G
80 PCLS
170 SCREEN1,1
175 FOR SS=15 TO 3 STEP-3
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90+C*PI/180
191 K=C*PI*180
200 X=INT(A-6+R*COS(K)):Y=INT(B-
8+R*SIN(C))
209 Q=RND(150)+100:L=RND(3):SOUN
DQ,L
210 PUT(X-40,Y)-(X-20,Y+25),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+25),S,PS
ET
215 IF Z=241 THEN FOR F=1 TO 400
0:NEXT:PCLS:GOTO170
220 NEXT Z,SS:PCLS:GOTO170
310 '***CHANGE LINE 209 TO Q=RND
(150)+100:L=RND(3):SOUNDQ,L FOR
THE CRISP NOTES OF THE ANTHEM,
'SQUEAKSQUAWK'.
```

Listing 2: TRACK

```
0 'GOOLIAN OLYMPICS---TRACK
10 'CREATED BY J.KOLAR, 1984
30 PMODE3:PCLS:PMODE4
40 A=126:B=90:R=72:PI=1.70
50 DIM S(13)
60 DRAW"BM0,0D24R2U3BU2U11F3R3D5
L3G3BD2E3R10F3D3R2U24L2D6BD2D11H
3L3U5R3E3BU2G3L3U3L4D3L3H3U5L2"
61 PAINT(10,7),1,1
70 GET(0,0)-(20,24),S,G
80 PCLS
170 SCREEN1,1
175 FOR SS=15 TO 3 STEP-3
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90+C*PI/180
200 X=INT(A-6+R*COS(C)):Y=INT(B-
8+R*SIN(C))
210 PUT(X-40,Y)-(X-20,Y+25),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+25),S,PS
ET
220 NEXT Z,SS:PCLS:GOTO170
```

Listing 3: LADDER

```
0 'GOOLIAN OLYMPICS---LADDER
10 'CREATED BY J.KOLAR, 1984
30 PMODE3:PCLS:PMODE4
40 A=126:B=90:R=72:PI=1.70
50 DIM S(13)
60 DRAW"BM0,0D24R2U3BU2U11F3R3D5
L3G3BD2E3R10F3D3R2U24L2D6BD2D11H
3L3U5R3E3BU2G3L3U3L4D3L3H3U5L2"
61 PAINT(10,7),1,1
70 GET(0,0)-(20,24),S,G
80 PCLS
170 SCREEN1,1
175 FOR SS=15 TO 3 STEP-3
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90+C*PI/180
200 X=INT(A-6+R*COS(2)):Y=INT(B-
8+R*SIN(C))
210 PUT(X-40,Y)-(X-20,Y+25),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+25),S,PS
ET
212 PUT(X+100,Y)-(X+120,Y+25),S,
PSET
220 NEXT Z,SS:PCLS:GOTO170
```

Listing 4: TUGOFWAR

```
0 'GOOLIAN OLYMPICS---TUGOFWAR
10 'CREATED BY J.KOLAR, 1984
30 PMODE3:PCLS:PMODE4
40 A=126:B=25:R=72:PI=1.70
50 DIM S(13)
60 DRAW"BM0,0D24R2U3BU2U11F3R3D5
L3G3BD2E3R10F3D3R2U24L2D6BD2D11H
3L3U5R3E3BU2G3L3U3L4D3L3H3U5L2"
61 PAINT(10,7),1,1
70 GET(0,0)-(20,24),S,G
80 PCLS
170 SCREEN1,1
175 FOR SS=15 TO 3 STEP-3
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90+C*PI/180
200 X=INT(A-6+R*COS(C)):Y=INT(B-
8+R*SIN(2))
209 Q=RND(150):L=RND(2):SOUNDQ,L
210 PUT(X-40,Y)-(X-20,Y+25),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+25),S,PS
ET
220 NEXT Z,SS:PCLS:GOTO170
```

Listing 5: PARAJUMP

```
0 'GOOLIAN OLYMPICS---PARACHUTE
JUMP
10 ' CREATED BY J. KOLAR, 1984
20 '
30 PMODE3:PCLS:PMODE4
40 A=128:B=42:R=80:PI=3.14
41 Q=33
50 DIM S(13)
60 DRAW"BM0,8M+2,-3M+6,-4M+2,-2M
+2,2M+6,4M+2,3L20F10NE10NM-4,-10
NM+4,-10 LDL2DR2D3LDR4ULU3R2UL2U
L"
61 PAINT(9,5),1,1
62 PAINT(10,20),1,1
70 GET(0,0)-(20,24),S,G
80 PCLS
170 SCREEN1,1
171 LINE(0,32)-(255,32),PSET
172 LINE(0,134)-(255,191),PSET,B
F
175 FOR SS=30 TO 30 STEP30
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90+C*PI/180
195 K=C*PI/-90
200 X=INT(A-6+R*COS(K)):Y=INT(B-
8+R*SIN(K))
210 PUT(X-40,Y)-(X-20,Y+24),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+24),S,PS
ET
212 PUT(X+100,Y)-(X+120,Y+24),S,
PSET
220 NEXT Z,SS:FOR H=1 TO5000:NEX
T:PCLS:GOTO170
```

Listing 6: PARALIFT

```
0 'GOOLIAN OLYMPICS---PARACHUTE
LIFT
10 ' CREATED BY J. KOLAR, 1984
20 '
30 PMODE3:PCLS:PMODE4
40 A=128:B=122:R=80:PI=3.14
41 Q=33
50 DIM S(13),T(13)
60 DRAW"BM0,8M+2,-3M+6,-4M+2,-2M
+2,2M+6,4M+2,3L20F10NE10NM-4,-10
NM+4,-10 LDL2DR2D3LDR4ULU3R2UL2U
L"
61 PAINT(9,5),1,1
62 PAINT(10,20),1,1
70 GET(0,0)-(20,24),S,G
71 LINE(200,0)-(220,24),PSET
80 PCLS
170 SCREEN1,1
171 LINE(0,32)-(255,32),PSET
172 LINE(0,136)-(255,191),PSET,B
F
175 FOR SS=30 TO 30 STEP30
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90-C*PI/180
195 K=C*PI/-90
200 X=INT(A-6+R*COS(K)):Y=INT(B-
8+R*SIN(K))
210 PUT(X-40,Y)-(X-20,Y+24),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+24),S,PS
ET
212 PUT(X+100,Y)-(X+120,Y+24),S,
PSET
216 PUT(X+30,Y)-(X+50,Y+24),T,PS
ET
220 NEXT Z,SS:FOR H=1TO2000:NEXT
:PCLS:GOTO170
```

Listing 7: BUCKLEUP

```
0 'GOOLIAN OLYMPICS---BUCKLE UP
THE SKY HOOK!
10 ' CREATED BY J. KOLAR, 1984
20 '
30 PMODE3:PCLS:PMODE4
40 A=128:B=122:R=80:PI=3.14
41 Q=33
50 DIM S(13),T(13)
60 DRAW"BM0,8M+2,-3M+6,-4M+2,-2M
+2,2M+6,4M+2,3L20F10NE10NM-4,-10
NM+4,-10 LDL2DR2D3LDR4ULU3R2UL2U
L"
61 PAINT(9,5),1,1
62 PAINT(10,20),1,1
70 GET(0,0)-(20,24),S,G
71 LINE(200,0)-(220,24),PSET
80 PCLS
170 SCREEN1,1
171 LINE(0,41)-(255,41),PSET
172 LINE(0,180)-(110,165),PSET:L
INE-(126,180),PSET:LINE-(255,180
),PSET:PAINT(0,181),1,1
175 FOR SS=30 TO 30 STEP30
180 FORZ=1TO2880 STEPSS:C=Z
190 C=90+C*PI/180
195 K=90-C*PI/-90
200 X=INT(A-6+R*COS(2)):Y=INT(B-
8+R*SIN(K))
210 PUT(X-40,Y)-(X-20,Y+24),S,PS
ET
211 PUT(X+30,Y)-(X+50,Y+24),S,PS
ET
212 PUT(X+100,Y)-(X+120,Y+24),S,
PSET
216 PUT(X+30,Y)-(X+50,Y+24),T,PS
ET
220 NEXT Z,SS
225 'PUT(X+30,Y)-(X+50,Y+24),S,P
SET
230 GOTO 230
```

64K MEMORY UPGRADE

by Brian Dougan

(Reprinted from CoCo Bug with the permission of the Author and the Editor).

With the trend to bigger and better programs for the Color Computer and the current lower prices of 64K chips the urge to go to 64K of RAM is effecting more and more people. If you are one of these and you are reasonably proficient with hand tools this article will try to guide you through the task.

First a word of warning!!

The dynamic RAM memory chips used for the CoCo are CMOS devices.

This type of IC has many advantages, however they also have the shortcoming that they are VERY sensitive to overvoltage damage.

This means that only a very small static electricity charge (far less than the ones you may have noticed generated when walking on carpet on very dry days) will leap from your fingers when you touch them and invisibly destroy them.

The answer is prevention. The chips you buy will come in an anti-static tube or inbedded in conductive black foam. Before you remove them you should ensure you are EARTHED - that is you are in contact with a good EARTH. This ensures any static electricity is discharged harmlessly before it can do damage. But you must be doubly careful to avoid any contact with 240v.

So do not have any 240v sources about while you EARTH yourself and handle the memory chips. Once they are in their sockets they are safe so then you can bring out the 240v soldering iron etc.

Types of CoCo:

TANDY have sold only "E", "F", and WHITE case COCO's in this country.

The "E" board generally is identified by the MEMORY SIZE BUTTON on the top right front of a GREY case. The "F" board is also in a GREY case but did not have a memory label, also the area immediately around the keyboard was changed to the GREY color of the case instead of the BLACK used on the "E". The WHITE case models sold recently i.e. the smaller case version uses a totally different type of memory chip and is physically very difficult to upgrade.

OK!! Hopefully you now are clear on which version you have and we can discuss the different approach to upgrading them.

"E" Board:

1. Unplug CoCo and turn it over, remove the 7 screws including the one under the warranty label (your warranty expired long ago)!

2. Turn it back over and remove the lid.

3. Unplug the keyboard and put it aside.

4. Remove capacitors:-C61,C31,C64,C35,C67,C45,C70, and C48 these are located between the memory chips at the front right corner of the board.

5. Shift the jumper block below C44 to the 16K/32K position. Shift jumper block between U8 and U4 to the 32K position. Shift the three jumper blocks near the keyboard connector to the 32K position.

6. Solder the middle and "LOW" pins near U29 together.

7. Solder the two pins to the left of C44 together.

8. Remove U29 from its socket (note the ident marks location to the top left). The pins are numbered from 1 to 7 anti-clockwise down one side and 8 to 14 up the other. Carefully bend pins 4,5, and 6 thru 180 degrees till they point straight up in the air. Replace U29 in its socket.

9. Similarly remove U11 and bend pin 5 up. Replace U11 in its socket.

10. Solder a wire between pin 6 and pin 8 (still in its socket) on U29.

11. Solder a wire between pin 4 of U29 and pin 5 of U11.

12. Solder a wire between pin 5 of U29 and TP1 in the back right corner of the board near the ROM PACK socket.

13. Remove all 240v from the area, EARTH yourself and gently remove the eight 16K chips. Replace them with the 64K chip making sure all pins fit cleanly into their socket and are not bent over.

14. Go back and double check the steps so far.

15. Remove your EARTH.

16. Replace keyboard, connect T.V. and turn it on. When it has warmed up, turn on the computer. If the screen fills with garbage turn off and check your workmanship. If SIGN-ON appears PRINT MEM should give 24728.

17. Replace lid ensuring short screws go in front holes.

"F" Board:

1. Steps 1,2,3 as for "E" board.

2. Remove C59-C66 and C70-C77 from near the memory chips.

3. There are 4 jumper blocks to shift to their 64K positions: Two are near the keyboard connector, one behind the memory chip on the right of the board, the last is to the left of U17 the left hand 40 pin PIA chip.

4. Steps 13 tto 18 as for the "E2" board.

White Long Case:

1. Steps 1,2,3, as for "E" board.

2. Remove C48-C52 from near the memory chips.

3. Remove solder straps P1-P4 from the 16K positions and rewire the 64K positions.

4. Steps 13 to 18 as for the "E" board.

TANDY ELECTRONICS DEALER (No. 9320)

TANDY COMPUTERS & ACCESSORIES

best prices!

FREE DELIVERY THROUGHT 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
47 High Street, Hastings, 3915.
(Dealer No. 9323.)
Ph. Hastings 059 79 1513

We produce Software

for

The CoCo and Tandy 1000

Our list of programs from

★	OWLS NEST SOFTWARE	U.S.A.
★	PAL CREATIONS	U.S.A.
★	CoCo CASSETTE (Monthly)	U.S.A.
★	PICOSOFT GAMES	U.S.A.
★	C.C.S.	AUSTRALIA
★	C.H.S.	AUSTRALIA

is too long to print here.

So, for a FREE CATALOGUE, call or write to

Computer Hut Software

21 WILLIAMS ST., BOWEN, QLD., 4805, AUSTRALIA — PHONE (077) 862220

Tandy ELECTRONICS

Greater Data Storage With a CoCo Thinline Disk Drive

Secondary Drive

ONLY

$449

Disk Drive #0

$599

FD-501 Color Thinline Disk Drive #0. Turn your Color Computer with extended BASIC into a complete disk system and store over 156,600 characters of data! Choose from our huge library of ready-to-run software or write your own sophisticated disk applications. Simply plug the drive into the Program Pak port or Multi-Pak interface. Color Disk Operating System includes 13.3cm double-density disk drive; Program Pak with disk operating program; cable; blank diskette; reference manual and operating instructions. 26-3129

NOW IN STOCK! **Secondary Thinline Disk Drive.** Combine both drives to create backup copies of valuable programs! A second drive also improves disk performance when you use one drive for storing data and the other for loading master programs. We recommend installation by our experienced technicians. 26-3130

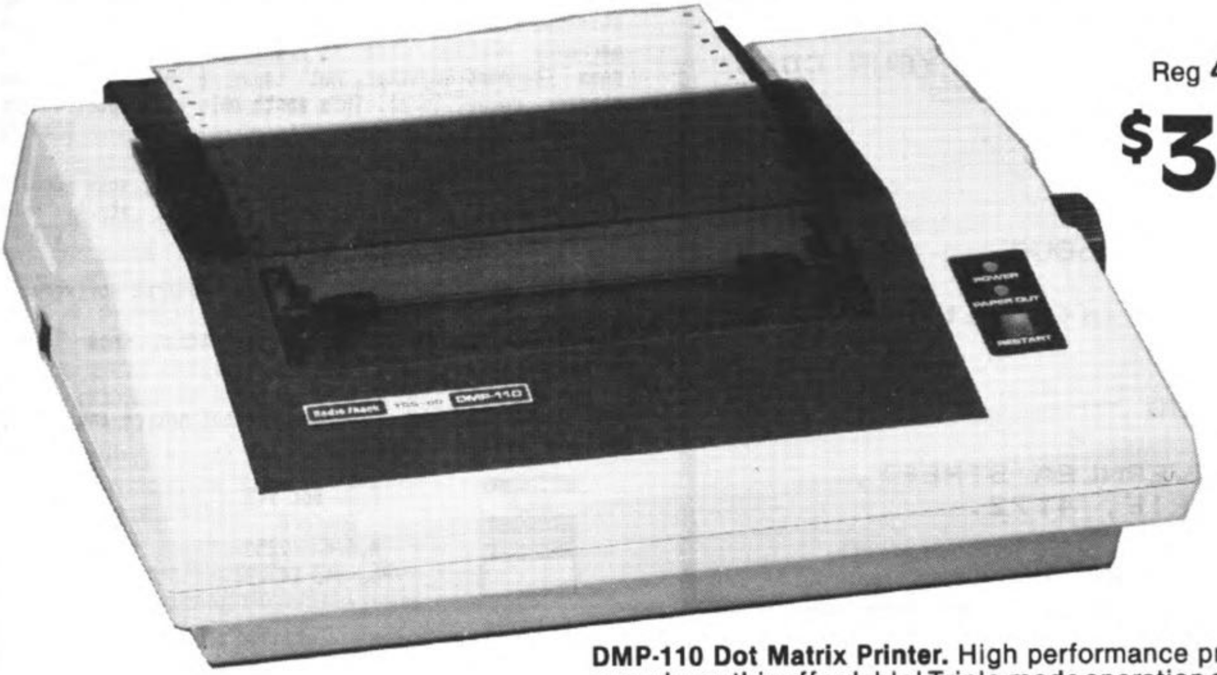
Over 50% Off! Tandy 13.3cm Double Density Diskette

Reg **6.95** **3^{45}**

Exclusive Offer to Color Computer Rainbow Readers!

Now there's no excuse for not having backups of all your data and program disks! This single-sided, 40-track diskette is made to exacting standards and is certified error-free. 26-305

SALE! Triple-Mode Personal Printer for Speed & Versatility



Reg **499.95**

$349

Save $150

DMP-110 Dot Matrix Printer. High performance printing has never been this affordable! Triple-mode operation offers word processing, data processing or dot-addressable graphics. Print proportionally spaced letters at 25 cpi and mono-spaced text in elite, standard or condensed at 50 cpi. Also prints italics, underline and microfonts. 26-1271

Printer and Serial Interface Cables

59⁹⁵

6⁹⁵

Standard Parallel Printer Cable. 34-pin edge to 36-pin. 26-1401

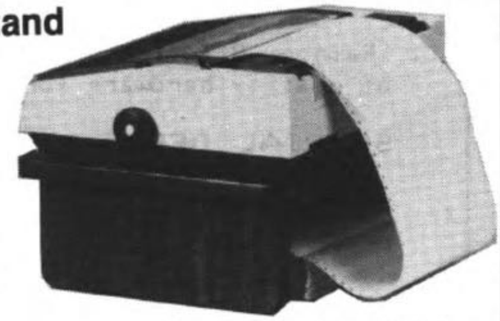
Printer and Modem Serial Interface Cable. 4-pin DIN to 4-pin DIN. 26-3020

Printer Stand

Save $10

Reg **59.95**

49⁹⁵



Printer and paper not included

Avoid paper-feeding problems and have easy access to your printer. Lower shelf 9.5cm deep. 26-1360

WE SERVICE WHAT WE SELL!

Tandy ELECTRONICS

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

Nearly 350 Stores Australia-Wide

Available from
350 Stores Australiawide
including Tandy Computer Centres

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

GRAPHICS

16K ECB

DYNAMIC COMPUTER GRAPHICS

By Charles W. Dold

A fascinating aspect of computer graphics is the ability to present a three-dimensional figure on the screen and then cause it to rotate about the pitch (X), yaw (Y) and roll (Z) axis of the figure. Such a presentation is possible with a relatively modest home computer if one is willing to accept some reduction in the speed and smoothness of motion and complexity of the figure. To demonstrate the possibility, let us first derive an algorithm to describe the task and the overall "plan of attack."

Assume the body is located in a coordinate system formed by three lines 'X', 'Y' and 'Z' passing through a point zero (origin), each line perpendicular to the other two (Figure 1).

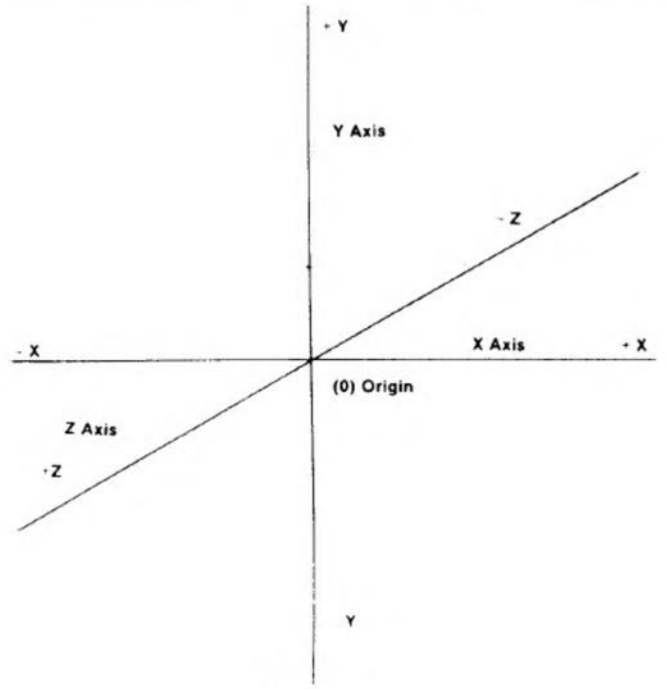


Figure 1
Coordinate System

The 'X' axis is horizontal with the plus values to the right of the origin and the negative values to the left of the origin. The 'Y' axis is vertical with the plus values upward and the minus values downward from the origin. The 'Z' axis is at right angles to the other two and, in relation to the computer screen, would project directly forward for plus values and behind the screen for minus values. The body is placed in this coordinate system such that the center of gravity of the body is located at the origin of the system.

The initial location of the body is established in the coordinate system by assigning the 'X', 'Y' and 'Z' coordinates of each corner of the body. First, let us look at a two-dimensional plane figure for simplicity. Consider a square located in the X,Y plane with its center located at the origin of the X,Y axis. (Since the square is a symmetrical figure, the center of the figure is also its center of gravity.) The location of the corners of the square are (X_1Y_1), (X_2Y_2), (X_3Y_3) and (X_4Y_4).

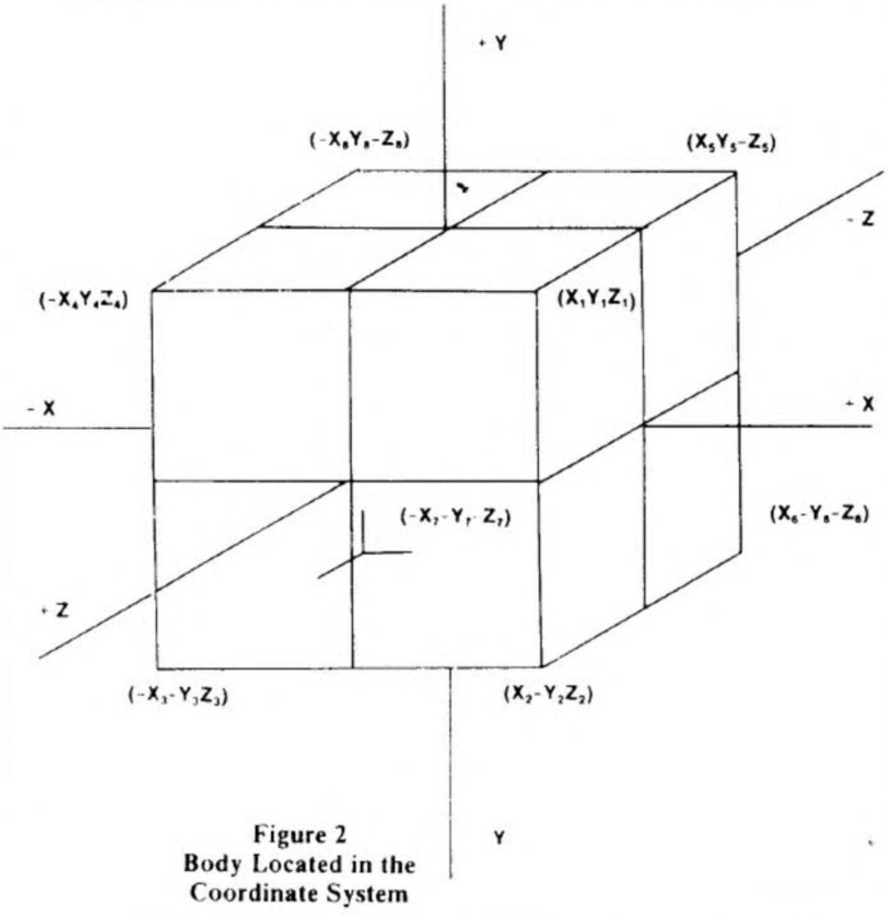


Figure 2
Body Located in the
Coordinate System

Assuming that X_1Y_1 is the upper right corner, progressing clockwise the signs of the corner coordinates are (X_1Y_1), (X_2-Y_2), $(-X_3-Y_3)$ and $(-X_4Y_4)$. Now if the square is rotated counter-clockwise about the origin through an angle of Φ degrees, the new coordinates of the corners will be $(X'_1Y'_1)$, $(X'_2-Y'_2)$, $(-X'_3-Y'_3)$ and $(-X'_4Y'_4)$. Using the equation of rotation (from our high school analytical geometry course) the new coordinates for each corner can be calculated from the initial coordinates as follows:

$$X' = X\cos\Phi + Y\sin\Phi$$
$$Y' = -X\sin\Phi + Y\cos\Phi$$

(It is necessary to observe the signs of the coordinates as we substitute them in the equations above.)

Now if we draw lines connecting the new corner coordinates, we will have drawn the square in its new position, rotated Φ degrees from its initial position.

So much for the two-dimensional (X,Y plane) figure. Now we will add the 'Z' axis. Take the plane containing the coordinates of the original square and slide it forward along the 'Z' axis to a point 'Z.' The new coordinates of the corners are $(X_1Y_1Z_1)$, $(X_2-Y_2Z_2)$, $(-X_3-Y_3Z_3)$ and $(-X_4Y_4Z_4)$. Slide the plane in the opposite direction from the origin along the 'Z' axis to a point -Z.

Starting with the upper right corner and numbering clockwise, the corners in this plane are numbers 5, 6, 7 and 8. The coordinates of the corners lying in the plane are $(X_5Y_5-Z_5)$, $(X_6-Y_6-Z_6)$, $(-X_7-Y_7-Z_7)$ and $(-X_8Y_8-Z_8)$. If we connect all eight corners we have drawn a cube (Figure 2).

Since the 'Z' coordinate of a corner is measured along the 'Z' axis to a line perpendicular to the 'Z' axis and passing through the corner, rotation of the corner about the 'Z' axis does not change the 'Z' coordinate of the corner. Therefore, the new coordinates of the corner after rotation about the 'Z' axis through the angle Φ degrees is:

$$X' = X\cos\Phi + Y\sin\Phi$$
$$Y' = -X\sin\Phi + Y\cos\Phi$$
$$Z' = Z$$

Rotation about the 'X' axis does not change the 'X' coordinates, therefore:

$$Z'' = Z'\cos\Theta + Y'\sin\Theta$$
$$Y'' = -Z'\sin\Theta + Y'\cos\Theta$$
$$X'' = X'$$

Rotation about the 'Y' axis does not change the 'Y' coordinates, therefore:

$$Z''' = Z''\cos\Psi + X''\sin\Psi$$

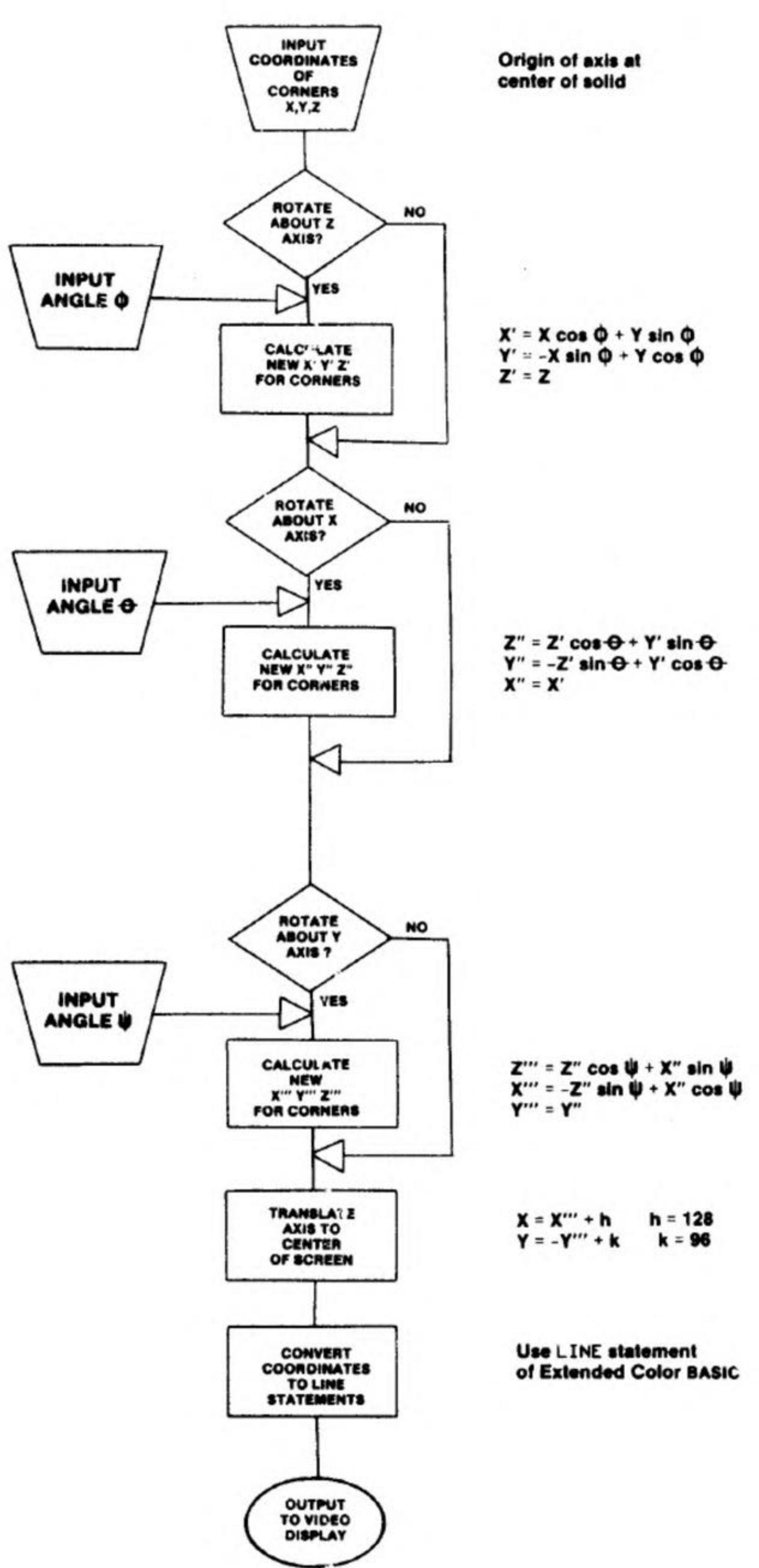

INPUT COORDINATES OF CORNERS X,Y,Z
Origin of axis at center of solid
ROTATE ABOUT Z AXIS?
NO
YES
INPUT ANGLE Φ
CALCULATE NEW X' Y' Z' FOR CORNERS
X' = X cos Φ + Y sin Φ
Y' = -X sin Φ + Y cos Φ
Z' = Z
ROTATE ABOUT X AXIS?
NO
YES
INPUT ANGLE Θ
CALCULATE NEW X'' Y'' Z'' FOR CORNERS
Z'' = Z' cos Θ + Y' sin Θ
Y'' = -Z' sin Θ + Y' cos Θ
X'' = X'
ROTATE ABOUT Y AXIS ?
NO
YES
INPUT ANGLE Ψ
CALCULATE NEW X''' Y''' Z''' FOR CORNERS
Z''' = Z'' cos Ψ + X'' sin Ψ
X''' = -Z'' sin Ψ + X'' cos Ψ
Y''' = Y''
TRANSLATE AXIS TO CENTER OF SCREEN
X = X''' + h   h = 128
Y = -Y''' + k   k = 96
CONVERT COORDINATES TO LINE STATEMENTS
Use LINE statement of Extended Color BASIC
OUTPUT TO VIDEO DISPLAY
STOP

Figure 3
Flow Chart

$X''' = -Z'' \sin \Psi + X'' \cos \Psi$
$Y''' = Y''$

New corner coordinates resulting from rotation about more than one axis are obtained, as shown, by solving the equations for rotation about each axis, in turn using the coordinates resulting from each previous set of calculations for substitution into the next following set of equations.

Figure 3 is a flow chart showing this sequence of operations in graphical form. Notice that after the three sets of equations of rotation have been solved, we are only interested in the final 'X' and 'Y' coordinates of the corners. This is because the computer screen contains only the X,Y plane and we are plotting the projection of the X,Y coordinates onto this plane. The 'Z' coordinates were necessary in the calculations to find the final X,Y coordinates.

The initial coordinates of the corners of the cube were taken with the origin of the coordinate system at the center of gravity (in the case of the cube, also the geometric center). This was done to simplify the equations of rotation.

The origin of the screen map of the Tandy Color Computer is the upper left-hand corner of the screen. If we were to plot the coordinates without correction, the cube would be plotted with its center located at the upper left corner of the computer screen. It is necessary to move the center of the cube to the right and down to center it on the screen. This is done with equations of translation (analytical geometry again):

$X = X + h$
$Y = -Y + k$

In the case of the TRS-80C, h = 128 and k = 96 (for a 256-by-192 graphics screen).

The TRS-80C graphics screen locations in the 'Y' direction are inverted. That is, the numbers increase in positive values as they go down the screen. In order to prevent the figure from being plotted in the inverted position the second equation is written as shown to compensate for the inversion.

Now we are able to plot the object at any angular attitude we desire. The next step is to make it rotate about one or more axes. Referring to the flow chart (Figure 3), we see that we input a fixed angle of rotation Φ about the 'Z' axis. Instead, we can input a variable angle 'S' which is made to vary between zero and 'M' degrees. To do this we use a FOR/NEXT loop, for example:

```
10 FOR S = 0 TO M STEP I
20 ⎫
30 ⎬ The equations of rotation
40 ⎭ about the 'Z' axis
50 NEXT S
```

The STEP I is added so the angular increments can be chosen. If we repeat for all eight corners of the cube, we will have calculated the new coordinates of all corners of the cube after rotation about the 'Z' axis in increments of 'I' degrees.

In like manner, we can calculate the coordinates of the corners after rotation of 2I, 3I, 4I, etc., until S = M. By substituting the equations of rotation about the 'X' or 'Y' axis in the FOR/NEXT loop, we can calculate the coordinates of the corners of the cube after rotation, in increments, about the 'X' or 'Y' axis. By having the computer plot the lines connecting each set of coordinates for each set of calculations, the cube will appear to rotate about its center.

As I mentioned in the beginning, there are some limitations to the quality of the presentation. The greatest is speed, the rate at which a new plot can follow the last one. When we consider that there are 64 calculations and 12 LINE statements to plot the cube in a single position it is easy to see why it takes so long.

A faster method that can be used for simple shapes with few corners (such as the cube) is to store the results from each set of calculations in a different array. After all calculations are complete and the results stored in arrays, the data can be retrieved from each array in turn and the figure defined by these data plotted. This eliminates the time for calculations from the sequence of plotting. The drawback of this method is that a great amount of memory is required.

A third method which conserves memory for complex shapes (but is somewhat slower than the previous methods) is to output the results of each set of calculations to tape and then, after all calculations are complete for all increments of rotation and recorded on tape, recall the data from tape one set at a time and plot the figures one following the other.

Now we have a plan of attack; let's look at the three programs for accomplishing it. We have some necessary inputs and options which are common to all three programs. These are prompted by lines 30 through 110. (Note: The line numbers refer to Listing 1. The statements are the same in Listings 2 and 3, but may have slightly different locations in the program.)

First, we must tell the computer how far to rotate the figure and in what increments or steps. Next, we are asked if we want a stroboscopic presentation. This means do we want to clear the screen after each plot (Line 620) or do we want to preserve all of the plots to get an effect similar to a stroboscopic camera picture? Then we have a choice of whether we want rotation held at a fixed angle about any of the three axes and, if so, at what fixed angle.

If a fixed angle other than zero degrees is selected for rotation about a particular axis, the incremented angle 'R' is saved (lines 190, 310, 430) and the fixed angle is substituted (lines 200, 320, 440). After the calculations for the axis are completed the angle 'R' is restored (lines 280, 400, 520). If we don't want any rotation about a particular axis we can specify rotation at zero degrees for that axis. If a fixed angle of zero degrees is selected for rotation about an axis, the coordinates for that axis are unchanged, therefore, the calculations for that axis are skipped (lines 180, 300, 420).

Lines 150 and 160 contain the initial location of the corner coordinates of the cubic solid. After a set of calculations is completed the data is plotted on the screen in the *GRAPHIC1* program, stored in an array in the *GRAPHIC2* program or stored on cassette tape in the *GRAPHIC3* program and 'S' is incremented for the next set of calculations. This will continue until S = M, the limit of rotation.

In the *GRAPHIC1* program (Listing 1) the FOR/NEXT loop (lines 130-760) includes the PLOT routine (lines 590-760), therefore the object is plotted after each set of calculations.

In the *GRAPHIC2* program (Listing 2) the FOR/NEXT loop (lines 130-570) does not include the plot routine (lines 600-760). Instead, the results of each set of calculations are put in an array in which the variables are labeled X(S + A) and Y(S + A) (Line 550). Thus for each value of 'S' a new array is established.

After arrays are filled for all values of 'S' the data is read out, one array at a time, in the FOR/NEXT loop (lines 580-770) which includes the plot routine. An array is read and the data plotted for each array in turn.

The *GRAPHIC3* program (Listing 3) is similar to the *GRAPHIC2* program, except the results of each set of calculations are recorded on cassette tape instead of stored in an array. To accomplish this, the cassette recorder is prepared to record (lines 130-140) before the FOR/NEXT loop is executed.

After completing a set of calculations, the line of communication to the tape recorder is opened (Line 610) and the results of the calculations recorded on tape, after which the line of communication is closed (Line 650). Following this the angle is incremented to the next 'S,' the cycle repeated until S = M and the data for all positions of the object are recorded.

We are now prompted to rewind the recorder and prepare to play back (lines 670-680). A new FOR/NEXT loop is established (lines 690-950) which, for each angle 'Z,' will open the communication line (Line 710), play back the data (lines 720-750) and plot the figure defined by that data (lines 770-930). After plotting the figure the communication line will be closed (Line 940) and the loop incremented to the next 'Z.' This will continue until Z = M and all the data from the tape is read and plotted.

At this point we are ready to run the program. Let us answer the prompts with the following inputs:

Limit of rotation	360 degrees
Rotation increments	10 degrees
Strobe presentation	No
Fixed rotation about 'Z' axis	Yes
Angle	20 degrees
Fixed rotation about 'X' axis	Yes
Angle	30 degrees
Fixed rotation about 'Y' axis	No

This is all that is required for the *GRAPHIC1* and *GRAPHIC2* programs. For the *GRAPHIC3* program you will be asked to set up the recorder with a blank tape to record and later, you will be asked to rewind and press Play.

It is immediately evident that both *GRAPHIC1* and *GRAPHIC2* programs are much faster than the *GRAPHIC3* program. The approximate run times with the inputs specified are as shown in Table 1.

Table 1

Program	Time (min:sec) Calculate	 Plot
GRAPHIC1	2:15	
GRAPHIC2	2:00	:15
GRAPHIC3	6:15	4:00

GRAPHIC1 and *GRAPHIC2* require the same overall run time, however, since the plot routine is separated from the calculations in *GRAPHIC2*, the plot time is by far the shortest.

One reason that *GRAPHIC1* and *GRAPHIC2* run time is less than *GRAPHIC3* is because of the use of the statement POKE 65495,0 to double the CPU rate. To demonstrate, try watching the flicker rate of the cursor as you POKE 65495,0 and ENTER. To get back to normal POKE 65494,0 and ENTER. You will notice that we return the CPU rate to normal at the end of the program.

If you should BREAK and not complete the program, it will be necessary to POKE the rate back to normal. This is especially important if you plan to CSAVE, CLOAD or input/output data to tape since these functions cannot be done at the high rate of the CPU. For this reason this scheme cannot be used in the *GRAPHIC3* program.

The programs can be modified to present other solids by substituting the coordinates of the corners of any solid for those of the cube as long as the conventions described at the beginning of this article are followed. If the number of corners is more or less than eight, it will be necessary to change all of the FOR/NEXT loops beginning with FOR A = 1 TO 8 to FOR A = 1 TO (X) where 'X' is the number of corners in the new solid.

```
140 ......181
410 ........34
590 ......201
END .....241
```

Listing 1: GRAPHIC1

```
1Ø '"GRAPHIC1" BY C.DOLD 1985
2Ø CLS:POKE65495,Ø
3Ø INPUT"LIMIT OF ROTATION IN DE
GREES";M
4Ø INPUT"ROTATION INCREMENTS IN
DEGREES";I
5Ø INPUT"IS STROBE PRESENTATION
DESIRED (Y)ES-(N)O";N$
6Ø INPUT"FIXED ROTATION ABOUT TH
E Z AXIS(Y)ES-(N)O";B$
7Ø IF B$="Y" THEN INPUT"ANGLE IN
 DEGREES";B
8Ø INPUT"FIXED ROTATION ABOUT TH
E X AXIS(Y)ES-(N)O";C$
9Ø IF C$="Y" THEN INPUT"ANGLE IN
 DEGREES";C
1ØØ INPUT"FIXED ROTATION ABOUT T
HE Y AXIS(Y)ES-(N)O";D$
11Ø IF D$="Y" THEN INPUT"ANGLE I
N DEGREES";D
12Ø DIMX(8):DIMY(8):DIMZ(8)
13Ø FOR S=Ø TO M STEP I:R=S/57.2
9577951
14Ø 'INITIAL COORDINATES FOR EAC
H CORNER OF THE SOLID
15Ø X(1)=5Ø:Y(1)=5Ø:Z(1)=5Ø:X(2)
=5Ø:Y(2)=-5Ø:Z(2)=5Ø:X(3)=-5Ø:Y(
3)=-5Ø:Z(3)=5Ø:X(4)=-5Ø:Y(4)=5Ø:
Z(4)=5Ø
16Ø X(5)=5Ø:Y(5)=5Ø:Z(5)=-5Ø:X(6
)=5Ø:Y(6)=-5Ø:Z(6)=-5Ø:X(7)=-5Ø:
Y(7)=-5Ø:Z(7)=-5Ø:X(8)=-5Ø:Y(8)=
5Ø:Z(8)=-5Ø
17Ø IF B$="N" THEN 21Ø
18Ø IF B=Ø THEN 29Ø
19Ø E=R
2ØØ IF B>Ø THEN R=B/57.2957791
21Ø FOR A=1 TO 8
22Ø 'CALCULATE NEW COORDINATES I
N THE X,Y PLANE
23Ø X=X(A)*COS(R)+Y(A)*SIN(R)
24Ø Y=-X(A)*SIN(R)+Y(A)*COS(R)
25Ø X(A)=X:Y(A)=Y
26Ø NEXT A
27Ø IF B$="N" THEN 29Ø
28Ø R=E
29Ø IF C$="N" THEN 33Ø
3ØØ IF C=Ø THEN 41Ø
```

```
31Ø E=R
32Ø IF C>Ø THEN R=C/57.29577951
33Ø FOR A=1 TO 8
34Ø 'CALCULATE NEW COORDINATES I
N THE Z,Y PLANE
35Ø Z=Z(A)*COS(R)+Y(A)*SIN(R)
36Ø Y=-Z(A)*SIN(R)+Y(A)*COS(R)
37Ø Y(A)=Y:Z(A)=Z
38Ø NEXT A
39Ø IF C$="N" THEN 41Ø
4ØØ R=E
41Ø IF D$="N" THEN 45Ø
42Ø IF D=Ø THEN 53Ø
43Ø E=R
44Ø IF D>Ø THEN R=D/57.29577951
45Ø FOR A=1 TO 8
46Ø 'CALCULATE NEW COORDINATES I
N THE Z,X PLANE
47Ø Z=Z(A)*COS(R)+X(A)*SIN(R)
48Ø X=-Z(A)*SIN(R)+X(A)*COS(R)
49Ø Z(A)=Z:X(A)=X
5ØØ NEXT A
51Ø IF D$="N" THEN 53Ø
52Ø R=E
53Ø FOR A=1 TO 8
54Ø 'TRANSLATION OF AXIS TO CENT
ER OF SCREEN
55Ø X(A)=X(A)+128
56Ø Y(A)=96-Y(A)
57Ø NEXT A
58Ø 'PLOT ROUTINE
59Ø PMODE 4,1
6ØØ IF R=Ø THEN 62Ø ELSE 61Ø
61Ø IF N$="Y" THEN 63Ø
62Ø PCLS
63Ø SCREEN 1,Ø
64Ø LINE (X(1),Y(1))-(X(2),Y(2))
,PSET
65Ø LINE -(X(3),Y(3)),PSET
66Ø LINE -(X(4),Y(4)),PSET
67Ø LINE -(X(1),Y(1)),PSET
68Ø LINE -(X(5),Y(5)),PSET
69Ø LINE -(X(6),Y(6)),PSET
7ØØ LINE -(X(7),Y(7)),PSET
71Ø LINE -(X(8),Y(8)),PSET
72Ø LINE -(X(5),Y(5)),PSET
73Ø LINE (X(6),Y(6))-(X(2),Y(2))
,PSET
74Ø LINE (X(3),Y(3))-(X(7),Y(7))
,PSET
75Ø LINE (X(4),Y(4))-(X(8),Y(8))
,PSET
76Ø NEXT S
77Ø POKE 65494,Ø
78Ø GOTO 78Ø
```

```
140 ....... 42
390 ....... 47
580 ...... 252
END ..... 251
```

Listing 2: GRAPHIC2

```
1Ø '"GRAPHIC2" BY C.DOLD 1985
2Ø CLS:POKE 65495,Ø
3Ø INPUT"LIMIT OF ROTATION IN DE
GREES";M
4Ø INPUT"ROTATION INCREMENTS IN
DEGREES";I
5Ø INPUT"IS STROBE PRESENTATION
DESIRED (Y)ES-(N)O";N$
6Ø INPUT"FIXED ROTATION ABOUT TH
E Z AXIS(Y)ES-(N)O";B$
7Ø IF B$="Y" THEN INPUT"ANGLE IN
 DEGREES";B
8Ø INPUT"FIXED ROTATION ABOUT TH
E X AXIS(Y)ES-(N)O";C$
9Ø IF C$="Y" THEN INPUT"ANGLE IN
 DEGREES";C
1ØØ INPUT"FIXED ROTATION ABOUT T
HE Y AXIS(Y)ES-(N)O";D$
11Ø IF D$="Y" THEN INPUT"ANGLE I
N DEGREES";D
12Ø DIM X(4ØØ):DIM Y(4ØØ):DIM Z(
4ØØ)
13Ø FOR S=Ø TO M STEP I:R=S/57.2
9577951
14Ø 'INITIAL COORDINATES FOR EAC
H CORNER OF THE SOLID
15Ø X(1)=5Ø:Y(1)=5Ø:Z(1)=5Ø:X(2)
=5Ø:Y(2)=-5Ø:Z(2)=5Ø:X(3)=-5Ø:Y(
3)=-5Ø:Z(3)=5Ø:X(4)=-5Ø:Y(4)=5Ø:
Z(4)=5Ø
16Ø X(5)=5Ø:Y(5)=5Ø:Z(5)=-5Ø:X(6
)=5Ø:Y(6)=-5Ø:Z(6)=-5Ø:X(7)=-5Ø:
Y(7)=-5Ø:Z(7)=-5Ø:X(8)=-5Ø:Y(8)=
5Ø:Z(8)=-5Ø
17Ø IF B$="N" THEN 21Ø
18Ø IF B=Ø THEN 29Ø
19Ø E=R
2ØØ IF B>Ø THEN R=B/57.2957791
21Ø FOR A=1 TO 8
22Ø 'CALCULATE NEW COORDINATES I
N THE X,Y PLANE
23Ø X=X(A)*COS(R)+Y(A)*SIN(R)
24Ø Y=-X(A)*SIN(R)+Y(A)*COS(R)
25Ø X(A)=X:Y(A)=Y
26Ø NEXT A
27Ø IF B$="N" THEN 29Ø
28Ø R=E
29Ø IF C$="N" THEN 33Ø
3ØØ IF C=Ø THEN 41Ø
31Ø E=R
32Ø IF C>Ø THEN R=C/57.29577951
33Ø FOR A=1 TO 8
34Ø 'CALCULATE NEW COORDINATES I
N THE Z,Y PLANE
35Ø Z=Z(A)*COS(R)+Y(A)*SIN(R)
36Ø Y=-Z(A)*SIN(R)+Y(A)*COS(R)
37Ø Y(A)=Y:Z(A)=Z
38Ø NEXT A
```

```
39Ø IF C$="N" THEN 41Ø
4ØØ R=E
41Ø IF D$="N" THEN 45Ø
42Ø IF D=Ø THEN 53Ø
43Ø E=R
44Ø IF D>Ø THEN R=D/57.29577951
45Ø FOR A=1 TO 8
46Ø 'CALCULATE NEW COORDINATES I
N THE Z,X PLANE
47Ø Z=Z(A)*COS(R)+X(A)*SIN(R)
48Ø X=-Z(A)*SIN(R)+X(A)*COS(R)
49Ø Z(A)=Z:X(A)=X
5ØØ NEXT A
51Ø IF D$="N" THEN 53Ø
52Ø R=E
53Ø FOR A=1 TO 8
54Ø 'TRANSLATION OF AXIS TO CENT
ER OF SCREEN. PUT IN ARRAYS
55Ø X(A)=X(A)+128:Y(A)=96-Y(A):X
(S+A)=X(A):Y(S+A)=Y(A)
56Ø NEXT A
57Ø NEXT S
58Ø FOR Z=Ø TO M STEP I
59Ø 'PLOT ROUTINE
6ØØ PMODE 4,1
61Ø IF Z=Ø THEN 63Ø ELSE 62Ø
62Ø IF N$="Y" THEN 64Ø
63Ø PCLS
64Ø SCREEN 1,Ø
65Ø LINE (X(1+Z),Y(1+Z))-(X(2+Z)
,Y(2+Z)),PSET
66Ø LINE -(X(3+Z),Y(3+Z)),PSET
67Ø LINE -(X(4+Z),Y(4+Z)),PSET
68Ø LINE -(X(1+Z),Y(1+Z)),PSET
69Ø LINE -(X(5+Z),Y(5+Z)),PSET
7ØØ LINE -(X(6+Z),Y(6+Z)),PSET
71Ø LINE -(X(7+Z),Y(7+Z)),PSET
72Ø LINE -(X(8+Z),Y(8+Z)),PSET
73Ø LINE -(X(5+Z),Y(5+Z)),PSET
74Ø LINE (X(6+Z),Y(6+Z))-(X(2+Z)
,Y(2+Z)),PSET
75Ø LINE (X(3+Z),Y(3+Z))-(X(7+Z)
,Y(7+Z)),PSET
76Ø LINE (X(4+Z),Y(4+Z))-(X(8+Z)
,Y(8+Z)),PSET
77Ø NEXT Z
78Ø POKE 65494,Ø
79Ø GOTO 79Ø
```

```
150 ......114
390 ......212
650 ......149
END .....127
```

Listing 3: GRAPHIC3

```
1Ø '"GRAPHIC3" BY C.DOLD 1985
2Ø CLS
3Ø INPUT"LIMIT OF ROTATION IN DE
GREES";M
4Ø INPUT"ROTATION INCREMENTS IN
DEGREES";I
5Ø INPUT"IS STROBE PRESENTATION
DESIRED (Y)ES-(N)O";N$
6Ø INPUT"FIXED ROTATION ABOUT TH
E Z AXIS(Y)ES-(N)O";B$
7Ø IF B$="Y" THEN INPUT"ANGLE IN
 DEGREES";B

8Ø INPUT"FIXED ROTATION ABOUT TH
E X AXIS(Y)ES-(N)O";C$
9Ø IF C$="Y" THEN INPUT"ANGLE IN
 DEGREES";C
1ØØ INPUT"FIXED ROTATION ABOUT T
HE Y AXIS(Y)ES-(N)O";D$
11Ø IF D$="Y" THEN INPUT"ANGLE I
N DEGREES";D
12Ø DIMX(8):DIMY(8):DIMZ(8)
13Ø CLS:PRINT"POSITION TAPE-PRES
S PLAY AND RECORD"
14Ø INPUT"PRESS<ENTER>WHEN READY
";R$
15Ø FOR S=Ø TO M STEP I:R=S/57.2
9577951
16Ø 'INITIAL COORDINATES FOR EAC
H CORNER OF THE SOLID
17Ø X(1)=5Ø:Y(1)=5Ø:Z(1)=5Ø:X(2)
=5Ø:Y(2)=-5Ø:Z(2)=5Ø:X(3)=-5Ø:Y(
3)=-5Ø:Z(3)=5Ø:X(4)=-5Ø:Y(4)=5Ø:
Z(4)=5Ø
18Ø X(5)=5Ø:Y(5)=5Ø:Z(5)=-5Ø:X(6
)=5Ø:Y(6)=-5Ø:Z(6)=-5Ø:X(7)=-5Ø:
Y(7)=-5Ø:Z(7)=-5Ø:X(8)=-5Ø:Y(8)=
5Ø:Z(8)=-5Ø
19Ø IF B$="N" THEN 23Ø
2ØØ IF B=Ø THEN 31Ø
21Ø E=R
22Ø IF B>Ø THEN R=B/57.2957791
23Ø FOR A=1 TO 8
24Ø 'CALCULATE NEW COORDINATES I
N THE X,Y PLANE
25Ø X=X(A)*COS(R)+Y(A)*SIN(R)
26Ø Y=-X(A)*SIN(R)+Y(A)*COS(R)
27Ø X(A)=X:Y(A)=Y
28Ø NEXT A
29Ø IF B$="N" THEN 31Ø
3ØØ R=E
31Ø IF C$="N" THEN 35Ø
32Ø IF C=Ø THEN 43Ø
33Ø E=R
34Ø IF C>Ø THEN R=C/57.29577951
35Ø FOR A=1 TO 8
36Ø 'CALCULATE NEW COORDINATES I
N THE Z,Y PLANE
37Ø Z=Z(A)*COS(R)+Y(A)*SIN(R)
38Ø Y=-Z(A)*SIN(R)+Y(A)*COS(R)
39Ø Y(A)=Y:Z(A)=Z
4ØØ NEXT A
41Ø IF C$="N" THEN 43Ø
42Ø R=E
```

```
430 IF D$="N" THEN 470
440 IF D=0 THEN 550
450 E=R
460 IF D>0 THEN R=D/57.29577951
470 FOR A=1 TO 8
480 'CALCULATE NEW COORDINATES I
N THE Z,X PLANE
490 Z=Z(A)*COS(R)+X(A)*SIN(R)
500 X=-Z(A)*SIN(R)+X(A)*COS(R)
510 Z(A)=Z:X(A)=X
520 NEXT A
530 IF D$="N" THEN 550
540 R=E
550 FOR A=1 TO 8
560 'TRANSLATION OF AXIS TO CENT
ER OF THE SCREEN
570 X(A)=X(A)+128
580 Y(A)=96-Y(A)
590 NEXT A
600 'STORE NEW COORDINATES ON TA
PE
610 OPEN"O",#-1,"ARRAY"
620 FOR A= 1TO 8
630 PRINT#-1,X(A),Y(A)
640 NEXT A
650 CLOSE#-1
660 NEXT S
670 PRINT"REWIND THE RECORDER AN
D PRESS PLAY"
680 INPUT"PRESS<ENTER>WHEN READY
";R$
690 FOR Z=0 TO M STEP I
```

```
700 'RECALL NEW COORDINATES FROM
 TAPE
710 OPEN"I",#-1,"ARRAY"
720 FOR A= 1 TO 8
730 IF EOF(-1) THEN 770
740 INPUT #-1,X(A),Y(A)
750 NEXT A
760 'PLOT ROUTINE
770 PMODE 4,1
780 IF Z=0 THEN 800 ELSE 790
790 IF N$="Y" THEN 810
800 PCLS
810 SCREEN 1,0
820 LINE (X(1),Y(1))-(X(2),Y(2))
,PSET
830 LINE -(X(3),Y(3)),PSET
840 LINE -(X(4),Y(4)),PSET
850 LINE -(X(1),Y(1)),PSET
860 LINE -(X(5),Y(5)),PSET
870 LINE -(X(6),Y(6)),PSET
880 LINE -(X(7),Y(7)),PSET
890 LINE -(X(8),Y(8)),PSET
900 LINE -(X(5),Y(5)),PSET
910 LINE (X(6),Y(6))-(X(2),Y(2))
,PSET
920 LINE (X(3),Y(3))-(X(7),Y(7))
,PSET
930 LINE (X(4),Y(4))-(X(8),Y(8))
,PSET
940 CLOSE#-1
950 NEXT Z
960 GOTO 960
```

In improving BOILER (going to double length arithmetic, etc.) I may find that speed demands a look-up table, but for now the trig routine will do. This is especially so since this column is supposed to be exploring FORTH. The most serious disadvantage of calculation methods, as mentioned by John, is that they are time-consuming.

However, lets look at the calculation method further. Study of appropriate maths textbooks shows that the series
X+((1/2)*(XY3/3))+((1*3/2*4)*(XY
5/5))+((1*3*5/2*4*6)*(XY7/7))+. .

Where (e.g.) XY5 is read X raised to the power 5.
Converges to ARCSIN X .

An interesting thing about the this method of calculating the value of arcsin (D1) is that one can make the calculation directly, rather than by employing ARCT(D1) as would probably be done in BASIC. See the arcsin routine in the CoCo manuals for further details.

Have a look at the way it has been calculated in the program and see if you can find a better way. Why not create a double length routine, send it in to the column and save me the trouble of writing it?

If your attempts to understand the FORTH program supplied this month are unsuccesful, please let me know where things stopped making sense, and I'll provide a detailed explanation. Good FORTH programmers will perhaps find cause to criticise. I can only put in a small plea for leniency on the basis of ignorance etc., and ask that you share your insights with me and others via this column.

I hope that this article has been of interest to you.
Please feel free to contact me on (07) 2080893.
Regards, John Poxon.

RAINBOW ON TAPE

Rainbow on Tape this month has an additional program called "Bigprint".

"Bigprint" by Michael Himowitz appeared originally in January 1984's Australasian Rainbow, and B.C. Burns of Kewdale, WA has made some changes which allow the program to fill the letter it is printing with the smaller standard size character, instead of the "X" the program originally used.

"Bigprint" is a nice program, and we regret our inability to reproduce it here.

Graham.

COMMLINK

64K ECB

A Question And Answer Repertoire

By R. Wayne Day

It's been a busy spring and it looks like it's getting busier, so let's dive right in with a review of some topics of conversation at the Irvine RAINBOWfest, a look at a way almost everyone can share graphics screens and some random notes.

Irvine Revisited

One of the highlights of speaking at the three seminars I participated in at Irvine was getting a chance to meet some of you, and I really enjoyed putting a face with some of the voices and names on your letters. To those who dropped by *The Color SIG*'s booth, we hope you enjoyed seeing what CompuServe is all about.

Special thanks go out to John Ross and Mike Ward, assistant SYSOPs for the SIG, and to Gerry Nolan, one of the SIG's old-time members, for manning the booth so well.

Questions, Questions

During and after the seminars, we had time to sit down and answer a few questions. Here are some of them that might be of interest to you:

Q: I'm thinking about buying a modem and wonder which one is "best"?

A: This is a question that has no hard and fast answer, but I can give you some general guidelines to follow.

Most modems on the market today work at least at 300 Baud (300 bits of information per second) and are compatible with the CoCo. Then, too, most terminal programs for the CoCo work excellently at 300 Baud, so that means the chances are good that your terminal program and your modem are compatible.

Many bulletin boards and information services offer a higher speed capability than 300 Baud, the most common being 1200 Baud. There are modems that work with both 300 and 1200 Baud, so you don't lose the lower speed capability when you upgrade your modem. Your terminal program, though, must be able to work at 1200 Baud as well for you to be able to use 1200 Baud modems. Again, remember your modem and terminal program must be compatible.

Q: Is a direct-connect modem better than an acoustically coupled modem?

A: Generally, the direct-connects are less prone to interference from noise in the room and usually work as well, if not better, than an acoustic modem.

Q: Are the cheaper modems any good?

A: I've had a chance to play with some of the less expensive modems being sold today and they seem to work very well over local telephone and AT&T long distance lines. I have had some trouble with using some of the supplementary telephone networks like MCI, Sprint, etc., but then again, I occasionally have some problems even with my "industry-standard" modem over those same lines.

Q: My terminal program says it works at 1200 Baud, but I've been having trouble with errors when I use it that fast. What's the problem?

A: The problem here may be twofold. First, the normal RS-232 port on the CoCo was designed in such a way that the programmer must place each individual bit to be sent over the RS-232 port into a memory address, one bit at a time. That is one reason why the CoCo's RS-232 port is referred to as a "bit-banger."

When using something like the Deluxe RS-232 ROM Pak, though, the programmer merely places the whole byte in a memory address, and the UART (Universal Asynchronous Receiver Transmitter) chip in the ROM Pak takes care of the housekeeping duties. At 300 Baud, there is usually more than enough time available for the program to send and receive data over the bit-banger.

At 1200 Baud, though, the transmission is sped up four times greater than at 300 Baud, and that's just a bit faster than most programs can handle perfectly.

One solution that many programs use is to force a "half-duplex" transmission at 1200 Baud, meaning you can only be receiving or sending data at one particular point in time, not both. That's normally OK, except when you want to interrupt what's coming in on the line, or when you need to respond rather quickly to something.

None of the programs I have personally used have been able to keep up with data at 1200 Baud reliably time after time when using the RS-232 port on the computer. The key here is "reliably." If you can afford an occasional error in your data, using 1200 Baud with the normal CoCo RS-232 port may be OK.

As more and more terminal programs that take advantage of the RS-232 ROM Pak become available, that problem will go away.

Q: I'm looking for a program like the guy used in the movie *War Games* to find BBS systems in my town (or county or area code or continent). Have you got one you'll share?

A: Nope. I've got a problem with auto-dialing programs such as that for two reasons.

First off, those programs seem to be used quite a bit at night, and even though your computer might not be bothered by a telephone that's answered by a human, I guarantee it doesn't work the same way around! Imagine what you would feel like if you were awakened at 2 a.m. by someone looking for a BBS system at random?

Secondly, BBSs aren't the only systems in the world that have dial-up terminals. And some of those dial-up numbers are definite no-no's, unless you're specifically authorized for them. Getting in and "hacking" at a dial-up is one of the things that has soiled the nickname of "hacker," and that's something I won't contribute to.

And More Questions

Many of the questions were about

CompuServe and *The Color SIG* in particular:

Q: My bills on CompuServe are way too much, and now someone is trying to tell me I can save money by using 1200 Baud. How is that possible when 300 Baud costs $6.25 at night and 1200 Baud costs $12.75?

A: Budgeting your time on CompuServe means budgeting your money, and it's easy to let either one get out of hand. If you are doing a lot of downloading or reading of messages on CompuServe, using 1200 Baud can be a money-saver for you, for with data coming at you four times faster than at 300 Baud, CompuServe charges you just a little bit over twice the normal rate.

If, on the other hand, you're sitting in the Conference mode chatting with other folks, 1200 Baud doesn't make any sense from a cost-effective standpoint.

What I recommend, if you have the capabilities, is to read the messages into a buffer at 1200 Baud, look at them and reply to them offline, then come back on at 1200 Baud to do your downloading, if you know what you want from the data libraries. Then use 300 Baud for exploring new areas on CIS, playing the online games and chatting in CB or Conference on the SIGs.

Q: I live in Canada and access CompuServe through Datapac, a Canadian telecommunications network. The CompuServe 'B' Protocol in *VIDTEX* doesn't work, and neither does *XMODEM*. What gives?

A: Datapac uses some of the same control characters that both CompuServe 'B' and *XMODEM* use, only for different purposes.

Without reconfiguring your Datapac line, it's not possible to successfully use either one, and so far, we've been successful in modifying Datapac for the CIS 'B' protocol only (see *DATAPK.TXT* in the DL4 database on CCSIG). This modification is not an "official" one, meaning that CompuServe's Customer Service office won't be able to help you if it doesn't work right the first time, but many folks have been able to use it successfully.

If any of our Canadian readers are aware of any method to get *XMODEM* working on Datapac, please let me know!

An alternative to consider, especially since Datapac charges are pretty hefty to begin with, would be to call one of the CompuServe nodes on the southern side of the border. Check with your local telephone company to see whether you might come out ahead doing that, and kill two birds with one stone.

Q: Will the RS-232 ROM Pak work on CompuServe?

A: Yes, it will — sort of. You can use the terminal functions on the Deluxe ROM Pak just as you can on any other information service and the print online functions work the same way.

As far as downloading, there's no way to use the RS-232 ROM Pak to download ASCII text files or machine language programs from CompuServe, since the program inside the ROM Pak assumes you're talking with another CoCo using the same software.

You can "fool" the ROM Pak into letting you download an ASCII BASIC program by going into the "Receive BASIC program" mode and letting the remote source dump the program to you. Then, when the "OK" prompt comes back, get back into BASIC where you have to save the program to tape immediately. After that, it's just a matter of re-EXECing the ROM Pak and continuing where you left off.

Q: What about *Color COMPAC*? Can I download with it?

A: The word I get from Tandy Customer Services is that due to a problem with the program itself, there's no way to get *COMPAC* to download anything from CIS. If anyone has been able to either patch *COMPAC*, or figure out how to get it to download something, again, please let me know.

Colorful Questions

One of the bigger hits at the Irvine RAINBOWfest was the introduction of *CoCo Max*, a new *Mac-Paint* clone of a graphics utility, which brought up another question heard frequently: "How can I put up a *CoCo Max* screen on my BBS where other folks can see it?"

The problem is that a full-sized *CoCo Max* screen is exactly twice the size of a "normal" CoCo graphics screen. It's like stacking one screen on top of another screen in one file.

Well, one of the creators of *PIXCMP.BAS* (or *PIXCMP.CC*), Art Flexser, who compressed a CoCo graphics screen into a BASIC program that can be uploaded to any BBS or CompuServe, has struck again; *MAXCMP.BAS* (or *MAXCMP.CC* on CompuServe) is the result.

Consider the problem: A *CoCo Max* screen is composed of binary data, residing (on a disk system) from address $E00 to a possible high address of $3DFF (that's decimal locations 3584 to 15871). For the CoCo user who doesn't have the capability of doing a protocol type transfer (i.e., *XMODEM*, DFT, CompuServe 'B,' etc.), the file must be converted into some form of ASCII data before he can upload or download it. *MAXCMP.BAS* looks at the data and converts the data into ASCII strings that, along with the onboard re-conversion routine, is written to disk as a stand-alone ASCII carrier for that binary data.

But, how do you look at the complete file, if a screen is twice the size of a normal CoCo screen? *MAXCMP.BAS* is smart! It looks at the file, and if it's too large to be viewed in one sitting, it "beeps" at you, alerting you to the fact that you can now use the up- and down-arrow keys to scroll through the picture to your heart's content.

Here are the detailed instructions.

1) Create the graphics screen (using almost any graphics utility that writes a screen to disk, with a starting address for the screen of $E00 — that includes *Graphicom*, *CoCo Max* and others).
2) `LOAD"MAXCMP.BAS"`
3) `RUN`
4) Give *MAXCMP* the filename of the screen you want to compress. *MAXCMP* assumes a default extension of /MAX, but you can override that easily by specifying another extension.
5) *MAXCMP* then loads the screen into memory and compresses it into ASCII data.
6) Give *MAXCMP* the filename of the new BASIC program you want to save. This is the program you'll actually upload to the BBS.
7) *MAXCMP* then writes the new BASIC program to disk in ASCII using the filename you gave it, and when it's done, you've got a BASIC program ready for uploading to your favorite BBS!

If your BBS needs a little "lift," get the folks started sharing those graphics pictures! It may be just the thing you need!

Of This 'n' That

One of the neater things about calling BBS systems around the country is seeing the specialization that occurs due to local factors. One of the more popular specializations I've seen in quite a while is evident on the BBS operated by Speech Systems in Batavia, Ill.

Their BBS, at (312) 879-6811, is chock full of music files that make the CoCo sound like a full-blown orchestra! Though they sell a music system for the CoCo, you don't have to be one of their customers to call and download music from the BBS, and thus, this BBS rates my "Must Call Award" for this month!

In April, we mentioned that Mike Randazzo of New Orleans has made available a couple of public domain terminal programs for the CoCo running under OS-9. Well, this month, we're quite pleased to relay that Mike, in between changing diapers for a new son, has also made available an *XMODEM* terminal program for the CoCo, and it too runs under OS-9. It's written entirely in BASIC09 and works quite well!

XMODEM.B09 runs at 300 Baud only and requires the use of the RS-232 ROM Pak as the /T1 device on your OS-9 system. You can download *XMODEM.B09* from Mike's Chalmette BBS at (504) 277-6926.

With this series of public domain programs, and the commercial terminal programs for OS-9 that have been appearing lately, it's good to see more and more OS-9 activity on the BBSs!

For those of you who have been looking for a public domain BBS system, Richard Duncan of West Memphis, Ark., may have the answer for you with his *COBBS* package. *COBBS* has been released into the public domain and the complete set of files can be downloaded from *The Color SIG* in the DL4 data library. See *COBBS.CAT* in DL4 for a list of the needed files and documentation. Richard is also making the package available outside of CompuServe, and you can make arrangements with him through COBBS #1 at (501) 735-5614.

Wrapping it up

Effective immediately, I can no longer be reached through MCI Mail, so please scratch that address from the list.

I may, however, be reached through TBBS Fort Worth at (817) 232-2087, through CompuServe (ID= 76703,376) or: Wayne Day, P.O. Box 79074, Fort Worth, TX 76179-0074. Please enclose a SASE if you desire a direct response.

New Listings

A/C	Phone	BBS Name	Location	Remarks
201	637-6286	Colorama of NW Jersey	<unknown>, NJ	
201	773-8265*	Meadowlands BBS	Meadow Park, NJ	
201	928-0949	Great American Coco	Jackson, NJ	
207	596-0556	Maine-Frame	Rockland, ME	evenings/weekends
305	731-6097	ComTech	Ft. Lauderdale, FL	HQ SYS
504	277-6926	Chalmette BBS	Chalmette, LA	OS9 Interest
602	344-8070*	Cactus Patch	Yuma, AZ	
614	475-0047	CocoNet Colorama	Gahanna, OH	
615	822-3752*	Fast Times	Hendersonville, TN	
619	368-3478	Inner Connection	<unknown>, CA	
619	437-1130	The Monitor	Coronado, CA	
707	538-7100	Archive BBS	Santa Rosa, CA	
714	534-5174*	Orange Co Color America	Huntington Beach, CA	
805	656-3746	Ventura BBS	Ventura, CA	
817	232-2087	TBBS Fort Worth	Fort Worth, TX	
818	996-1977	TRS-80 Country	Inglewood, CA	
919	425-7139	Color 80 #82	Fayettville, NC	
Deletions				
408	984-XXXX	Rainbow #5	San Jose, CA	Now Offline
619	474-8981	JARB / Coco Sig	San Diego, CA	Now Offline
Changes				
201	827-7815	PeopleLinks	Ogdensburg, NJ	now Sussex County BBS
213	244-XXXX	Fantasy Plaza	Burbank, CA	new number (818) 840-8252

		88	197
17	20	109	197
36	42	127	132
55	123	144	92
71	39	END	29

The listing: MAXCMP

```
1 '***MAXCOMP***'
2 '
3 'THIS PROGRAM TAKES A FILE SAV
ED FROM A HI-RES GRAPHICS
4 'SCREEN AND CREATES FROM IT A
RUNNABLE BASIC PROGRAM,
5 'SUITABLE FOR BBS UPLOADING, T
HAT RECREATES THE PICTURE FROM
6 'A COMPRESSED ASCII FORMAT AND
 SAVES THE BINARY OUTPUT TO
7 'DISK OR TAPE, ACCORDING TO WH
ETHER OR NOT A DISK CONTROLLER
8 'IS PRESENT.  MAXCOMP REQUIRES
 64K, DISK OR CASSETTE.
9 '
10 'MAXCOMP IS AN ENHANCED VERSI
ON OF PIXCOMP, BY MIKE WARD
11 'AND ART FLEXSER, WHICH HAS H
AD ITS CAPABILITIES EXTENDED TO
12 'ALLOW USE WITH FILES SUCH AS
 THOSE CREATED BY COCO MAX,
13 'WHICH MAY BE EITHER ONE OR T
WO PMODE 4 GRAPHICS SCREENS IN
14 'LENGTH.  WHEN A PROGRAM CREA
TED BY MAXCOMP IS RUN, A BEEP
15 'WILL SOUND AFTER THE PICTURE
 IS CREATED IF THE PICTURE IS
16 'LARGER THAN CAN BE DISPLAYED
 ON THE SCREEN.  THE BEEP
17 'INDICATES THAT THE UP AND DO
WN ARROW KEYS MAY BE USED TO
```

```
18 'SCROLL THE PICTURE UP AND DO
WN TO VIEW IT IN ITS ENTIRETY.
19 'PRESSING THE 'S' KEY WILL CA
USE A PROMPT FOR A FILENAME TO
20 'SAVE THE PICTURE FILE IN BIN
ARY FORMAT, SUITABLE FOR
21 'LOADING INTO A GRAPHICS UTIL
ITY.
22 '
23 'ART FLEXSER, MIAMI, MARCH 19
85
24 GOTO26
25 GOTO27
26 PCLEAR8:GOTO25
27 CLEAR800,&H7A00:GOTO31
28 C$=CHR$(34):GOTO30
29 C$=" "
30 N=N+1:L$=MID$(STR$(N),2)+C$+A
$:PRINT#U,L$:PRINT L$:RETURN
31 CLS:A$=CHR$(232):S$=STRING$(2
3,232):PRINT@67,S$:PRINT@199,S$
32 FORI=0TO2:PRINT@100+33*I,A$TA
B(26+I)A$:NEXT
33 PRINT@138,"M A X C O M P";:PR
INT@265,"BINARY TO ASCII"
34 PRINT@296,"PICTURE COMPRESSOR
":PRINT@392,"ARTHUR J. FLEXSER"
35 PRINT@431,"AND":PRINT@457,"MI
CHAEL D. WARD"
36 IFPEEK(&HC000)=68THENS=&HE00:
D=1:U=1 ELSE S=&H600:U=-1
37 FOR X=&H7A00 TO &H7A5D
38 READ H$:POKE X,VAL("&H"+H$):N
EXT
39 DATA BD,B3,ED,C3,0,3,ED,8C,58
,1A,50,7F,FF,DF,8E,80,0,10,AE
40 DATA 8D,5,E6,86,8,A7,8C,44,86
,6,A7,8C,3E,6F,8C,3D,A6,80,5F
41 DATA 48,59,6A,8C,33,27,16,6A,
8C,2F,26,F4,86,8,A7,8C,28,A6,80
42 DATA AC,8C,25,25,E8,63,8C,1F,
20,E3,CB,30,E7,A0,C6,6,E7,8C,12
43 DATA 5F,6D,8C,10,27,DB,31,3F,
1F,20,7F,FF,DE,1C,AF,7E,B4,F4
44 FOR X=&H7B00 TO &H7C5D
45 READ H$:POKE X,VAL("&H"+H$):N
EXT
46 DATA EC,8D,4,FA,C3,0,1,ED,8D,
1,55,83,0,20,ED,8D,1,50
47 DATA 83,0,20,ED,8D,1,4B,EC,8D
,4,E1,A3,8D,4,DB,83,0,21
48 DATA 43,53,ED,8D,1,0,ED,8D,1,
B,CB,20,ED,8D,1,14,1A,50
49 DATA 7F,FF,DF,8E,0,0,4F,5F,31
,8D,1,24,AF,A1,A7,A0,4C,5A
50 DATA 26,F8,AE,8D,4,AE,31,8D,1
,14,86,3,E6,80,3D,34,20,31
51 DATA AB,EE,A4,33,41,EF,A4,35,
20,AC,8D,0,F9,25,E9,31,8D,0
52 DATA F9,C6,FF,AE,A4,A6,22,AC,
23,25,4,AE,23,A6,25,31,23,5A
53 DATA 26,F3,B7,80,0,6F,8D,0,D8
,AE,8D,4,71,10,8E,80,1,A6
54 DATA 84,17,0,8A,17,0,B4,27,12
,B1,80,0,26,9,F6,80,0,E7
55 DATA A0,C6,1,E7,A0,A7,A4,20,6
9,A1,84,26,52,34,2,17,0,79
56 DATA A1,E4,35,2,26,47,34,2,17
,0,7D,A1,E4,35,2,26,3C,C6
57 DATA 1,E7,8D,0,93,C6,FF,A1,84
,26,12,6C,8D,0,89,17,0,46
58 DATA 17,0,70,26,15,E1,8D,0,7D
,22,EA,F6,80,0,E7,A0,E6,8D
59 DATA 0,72,E7,A0,A7,A0,20,9D,F
6,80,0,E7,A0,E6,8D,0,63,E7
60 DATA A0,A7,A4,20,13,B1,80,0,2
6,9,F6,80,0,E7,A0,C6,1,E7
61 DATA A0,A7,A0,16,FF,7B,7F,FF,
DE,1C,AF,1F,20,7E,B4,F4,AC,8D
62 DATA 0,40,24,4,30,88,20,39,30
,89,E8,21,39,AC,8D,0,31,24
63 DATA 4,A6,88,20,39,A6,89,E8,2
1,39,AC,8D,0,24,24,4,A6,88
64 DATA 40,39,A6,89,E8,41,39,6D,
8D,0,10,26,C,AC,8D,3,A9,26
65 DATA 4,6C,8D,0,4,1A,4,39
66 DEFUSR0=&H7B00:DEFUSR1=&H7A00
67 CLS:PRINT@64,"FILE NAME OF PI
CTURE":IF D THEN E$="/MAX"
68 '/MAX IN ABOVE LINE CAN BE CH
ANGED TO /BIN IF PREFERRED
69 IF D THENPRINT@128,"(DEFAULT
EXTENSION IS ";E$;")":PRINT@85
70 LINEINPUT"TO COMPRESS: ";F$:I
F D THEN Y$="/BAS":GOSUB156
71 IF D AND F$="" THEN 67
72 PMODE4,5:PCLS1:PMODE4,1:PCLS1
:SCREEN1,1
73 IFD THEN LOADM G$ ELSE CLOADM
 F$:GOTO76
74 OPEN "D",1,G$,3:GET #1,1:CLOS
E 1
75 L=256*PEEK(&H98A)+PEEK(&H98B)
:GOTO77
76 L=256*PEEK(&H7E)+PEEK(&H7F)-&
H600
77 NS=INT((L-1)/6144)+1:L=6144*N
S
78 GS=S/256:GE=GS+&H18*NS-1
79 POKE&H7FFC,GS:POKE&H7FFD,0:PO
KE&H7FFE,GE:POKE&H7FFF,&HFF
80 E=USR0(0):E2=USR1(E):IF E<0 T
HEN E=E+65536
81 PRINT:PRINTUSING"#####";L;:PR
INT" BYTES ORIGINALLY"
82 PRINTUSING"#####";E-&H8000+1;
83 PRINT" BYTES AFTER COMPRESSIO
N"
84 PRINTUSING"#####";E2-S;:PRINT
" BYTES CONVERTED TO ASCII"
85 PRINT:PRINT@320,"FILENAME FOR
 ASCII PICTURE"
86 IF F$<>""THEN PRINT"OR <ENTER
> TO USE: "TAB(8)F$+Y$+DN$
87 PRINTSTRING$(17,"=");"> ";:LI
NEINPUT G$
88 IFG$=""THENG$=F$+Y$+DN$:IF F$
=""THEN85
89 IF D THEN E$="/BAS":F$=G$:GOS
UB 156
90 PRINT:PRINT"SAVING ";G$:PRINT
91 OPEN "O",#U,G$
92 IF NS<>1 THEN LN$="28" ELSE L
N$="24"
93 AD=S:EC=59:A$="CLS:CLEAR200,&
H7F00:GOTO"+LN$:GOSUB29
94 IF NS=1 THEN 98
95 A$="PRINT"+CHR$(34)+"USE UP A
ND DOWN ARROW TO SCROLL"
96 A$=A$+CHR$(34)+":PRINT"+CHR$(
34)+"THE SCREEN."+CHR$(34)
97 GOSUB 29
98 A$="PRINT:PRINT"+CHR$(34)+"HI
T 'S' TO SAVE PICTURE AS A"
99 A$=A$+CHR$(34)+":PRINT"+CHR$(
34)+"BINARY FILE"+CHR$(34)
100 GOSUB29:A$="S=6:E=&H"+HEX$(G
E-8*D)+":IF PEEK"
101 A$=A$+"(&HC000)=&H44 THEN D=
1:S=S+8:E=E+8":GOSUB29
102 A$="POKE&H7FFC,S:POKE&H7FFD,
0:POKE&H7FFE,E:POKE&H7FFF,&HFF"
103 IF NS<>1 THEN A$=A$+":V=S"
104 GOSUB 29:A$="FORI=&H7F00 TO&
H7FB7:READ H$:POKE I,VAL("
105 A$=A$+CHR$(34)+"&H"+CHR$(34)
+"+H$):NEXT"
106 IF NS<>1 THEN A$=A$+":U=S+24
"
107 GOSUB 29
108 A$="DATA EC,8D,0,FA,83,0,1F,
ED,8D,0,AF,A3,8D,0,ED,43,50,5C"
109 GOSUB29:A$="DATA ED,8D,0,8E,
1A,50,7F,FF,DF,9E,33,30,6,10,"
110 A$=A$+"8E,80,0,86":GOSUB29:A
$="DATA 8,A7,8C,3A,86,6,A7,8C"
111 A$=A$+",34,A6,80,80,30,48,48
,48,59,6A":GOSUB29:A$="DATA "
112 A$=A$+"8C,29,27,E,6A,8C,25,2
6,F4,E7,A0,C6,8,E7,8C,1C,20,EB"
113 GOSUB29:A$="DATA 86,6,A7,8C,
14,A6,80,26,A,A6,4,81,22,26,C"
114 A$=A$+",30,5,A6":GOSUB29:A$=
"DATA 80,80,30,48,48,20,D9,0,"
115 A$=A$+"0,8E,80,0,10,AE,8D,0,
91,A6":GOSUB29:A$="DATA 80,A7"
116 A$=A$+",8C,48,6F,8C,46,A6,80
,A1,8C,40,26,F,E6,80,A6,80,A7"
117 GOSUB29:A$="DATA A4,8D,15,8D
,23,5A,26,F7,20,4,A7,A4,8D,A,"
```

```
118 A$=A$+"8D,18,27,E2,7F":GOSUB
29:A$="DATA FF,DE,1C,AF,39,10"
119 A$=A$+",AC,8D,0,1E,24,4,31,A
8,20,39,31,A9,E8":GOSUB29
120 A$="DATA 21,39,6D,8C,F,26,B,
10,AC,8C,4E,26,3,6C,8C,4,1A,4,"
121 A$=A$+"39":GOSUB29
122 A$="PMODE4:PCLS1:SCREEN1,1:R
EADZ:EXEC&H7F00:CLS"
123 IF NS<>1 THEN A$=A$+":SOUND
200,1"
124 GOSUB29
125 IF NS=1 THEN X$="<>251 THEN
17" ELSE X$="=251 THEN 22"
126 A$="P=PEEK(&H155):IF P"+X$:G
OSUB29
127 IF NS=1 THEN 132
128 A$="IF P=247 THEN V=V+1:POKE
&H155,&HFF:IF V>U THEN V=U"
129 GOSUB29:A$="IF PEEK(&H156)=2
47 THEN V=V-1:POKE&H156,&HFF"
130 A$=A$+":IF V<S THEN V=S":GOS
UB29
131 A$="POKE&HBA,V:SCREEN1:GOTO1
8":GOSUB 29
132 A$="IFD THEN PRINT@64,"+CHR$
(34)+"(DEFAULT EXTENSION IS "
133 A$=A$+"/MAX)"+CHR$(34)+":PRI
NT@0,"+CHR$(34)+CHR$(34)+";"
134 GOSUB29
135 A$="LINEINPUT"+CHR$(34)+"FIL
ENAME FOR PICTURE: "
136 A$=A$+CHR$(34)+";F$:IFF$="+C
HR$(34)+CHR$(34)+"THEN"
137 A$=A$+STR$(N):GOSUB29:A$="P=
INSTR(F$,"+CHR$(34)+":"
138 A$=A$+CHR$(34)+"):IFP>2 THEN
 F$=MID$(F$,P+1)+"+CHR$(34)
139 A$=A$+":"+CHR$(34)+"+LEFT$(F
$,P-1)":GOSUB29
140 A$="IFD AND INSTR(F$,"+CHR$(
34)+"/"+CHR$(34)+")=0 AND "
141 A$=A$+"INSTR(F$,"+CHR$(34)+"
."+CHR$(34)+")=0THEN F$=F$+"
142 A$=A$+CHR$(34)+"/MAX"+CHR$(3
4):GOSUB29
143 A$="IFD THEN SAVEM F$,&HE00,
256*(E+1)-1,&HA027:END"
144 GOSUB29:A$="CSAVEM F$,&H600,
256*(E+1)-1,&HA027:END"
145 GOSUB29:A$="PCLEAR"+STR$(4*N
S)+":GOTO2":GOSUB29
146 A$="DATA1":GOSUB29
147 C1=INT((E2-S)/EC)
148 C2=(E2-S)-(C1*EC):N=99
149 V=VARPTR(A$):V1=V+2:V2=V+3:P
OKEV,EC
150 FOR I=1 TO C1
151 GOSUB154:AD=AD+EC:NEXT:IFC2=
0THEN153
152 POKEV,C2:GOSUB154
153 CLOSE:PCLEAR4:CLEAR200,&H7FF
F:END
154 MS=INT(AD/256):LS=AD-256*MS:
POKEV1,MS:POKEV2,LS:GOSUB28
155 RETURN
156 P=INSTR(2,F$,":"):IF P=0 THE
N 159
157 IF P=2 THEN DN$=":"+LEFT$(F$
,1):F$=MID$(F$,3):GOTO159
158 DN$=MID$(F$,P):F$=LEFT$(F$,P
-1)
159 P=INSTR(2,F$,"/"):IFP=0 THEN
 P=INSTR(2,F$,".")
160 IF P<>0 THEN E$=MID$(F$,P):F
$=LEFT$(F$,P-1)
161 G$=F$+E$+DN$:RETURN
```

TURN OF THE SCREW

Look, Ma No Switch!

By Tony DiStefano

Before I get into this month's project, I want to thank my readers for being so good about calling me only on Monday nights. I have had many interesting calls, and not all on problems relating to my projects, either. Some just call me to discuss theory and hardware.

For all new readers to this column, let me explain what I am talking about. I have set aside Monday nights for people to call me about problems they might encounter in putting my projects together. My number [in Canada] is (514) 473-4910. If by chance you want to write me a letter, by all means do, but if you want a written answer please include a self-addressed, stamped envelope, otherwise I'll take it as just a point of interest. But please keep the calls to Monday nights.

The next thing on the agenda is a piece of software/hardware I had the pleasure of trying — the *CoCo Max* drawing package. The software part is great. The hardware part is used to read the joystick (or in this case, a mouse) more precisely than the CoCo's internal joystick connection.

In my case, it was no problem to connect this ROM pack-looking adapter to CoCo; I have a Radio Shack Multi-Pak Interface. I just plug it in and away we go. This, however, might be a problem to users who have a disk drive and no Multi-Pak Interface: they both plug into the Expansion Slot on the side of the CoCo.

In order to have both the disk drive controller and the joystick connected together without a Multi-Pak, you need an adapter. These adapters (better known as 'Y' adapters), don't come cheap. I did an article on how to make a 'Y' adapter for your CoCo in the July 1983 isssue ("Build A 'Y' Adapter For Your Disk Controller," Page 176). If you don't have it, call up RAINBOW to get a back issue. This adapter will work with the *CoCo Max* and isn't nearly as expensive.

On with the Project

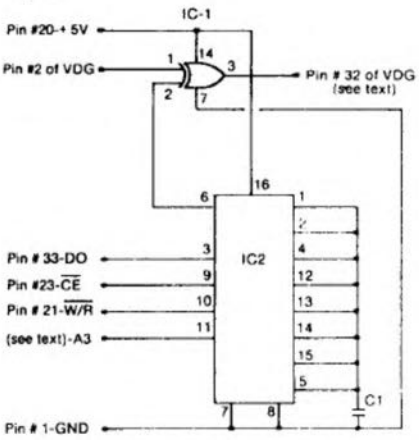
Once upon a time, long, long ago in a place far, far away I did a project on how to get inverse video on the CoCo. For those of you who are not familiar with the term "inverse video," let me explain: When you turn on the CoCo, you are greeted with a green square on the screen with black letters inside — the "normal" screen. When you enter in lowercase letters, you get this black square with a green letter in it. That is how you can tell it is lowercase.

Anyway, the project was a modification so that the computer would show a totally black screen with green uppercase letters. The lowercase letters were black with a green square. Whatever was green before was now black and whatever was black before was now green. Consequently, the term "inverse video."

Included with the circuit was a switch so you could switch between inverse video and normal video. I also said if there was enough interest I would come up with a circuit that would let you do the switching (the part of the physical switch) in software. There was; I did; here it is.

This month, I will show you the circuit and how to connect it so a simple POKE command will switch you between normal and inverse video. If you have an EPROM burner and know how to modify the DOS, you can have it built into the DOS. In order to build this circuit, you will need the standard kit builder's paraphernalia. You must also be unafraid of digging into your CoCo. As far as I know, this circuit will work on any version of the CoCo family.

Figure 1



All wires connect to the PIA except where noted (see text)

The first part, as usual, is the circuit. Find in Figure 1 the circuit for this electronic switch. It consists of two chips and one capacitor. In order to understand how this circuit works, you must first know a little about the chips I used. The first chip is the 74LS86. This chip is a quadruple 2-input exclusive-or. Exclusive means one or the other but not both. This short truth table will help explain.

Input A	Input B	Output
L	L	L
L	H	H
H	L	H
H	H	L

H = High level (1) L = Low level (0)

In my circuit I used the 'B' input of one of the four gates as the inverter control. When 'B' is high the output is inverted with respect to the input 'A'. When 'B' is low, the output is the same with respect to the input 'A'. This is nothing new and it has been done before. A 74LS86 is almost always used for this purpose. If you were to put a switch here, you would have an inverted controller, switchable by hand. My circuit goes one further.

The next chip is a little more tricky. What is required from this part of the circuit is a type of switch action with memory, which means memory mapping (remember last month?).

This puts theory to good use. Inside the CoCo there are two possible areas we can use: either the keyboard PIA (MC6821 or MC6822) or the DAC (Digital to Analog Convertor) PIA. They are mapped at $FF00 (65280) and $FF20 (65312), respectively. (I'll go into the differences between the two later.) Both of these I/O areas are 32 bytes long. The functions these I/O areas control (the PIA) require only four bytes, and the other 28 bytes are memory mirrored. What this chip does is decode this area into two halves. This becomes either $FF10 (65296) or $FF30 (65328), depending on what PIA you hook it up to.

In order to do this, we need four signals. The first is the chip enable from the PIA that selects the 32 bytes. The next is the Read/Write line, to make the memory mapped byte a write-only byte. The next signal needed is address line #3, which is the address line needed to get us the two halves of the area. The final signal needed is a data line so we can write a signal to control the 74LS86 and latch it. All this adds up to a one-bit latch that can be changed by software; our one output needed to feed the 'B' input (control) of the 74LS86.

The chip used is a 74LS151. This chip is a one-of-eight Data Selector/Multiplexer. Although the chip is not known for its latching capabilities, my good friend and co-worker Larry Callahan and I worked out a way to latch the output with the input by using a small capacitor. By feeding the output to all of the inputs but one with a capacitor, it acted as a latch with the last input as a control. We then used the inverting output as a latched bit to control the input 'B' of the 74LS86.

When a zero is written to chip it is remembered by the capacitor and other inputs. When a 1 is written to the same location, it is remembered as well. That is all there is to it. For more information on these and all the "74LS" family of chips, refer to *The TTL Data Book* by Texas Instruments. By the way, the value of the capacitor is a mere 100 PF at 50 volts.

To construct this circuit, I used a little piece of perf board one-half inch by two inches and stuck it on top of the PIA using double-sided tape. I ran the wires along the sides to the proper pins of the PIA and the VDG (MC6845). I soldered directly to the pins of the PIA and VDG, though, some like to use a socket so as not to damage the components themselves.

All of the wires just need to be connected but, there is one pin that has to be lifted out of its socket; that is pin #32 of the VDG. If your VDG is not soldered in, just remove it, bend the pin up on an angle, replace the chip back in its socket, and then solder your wire to the lifted pin. If your VDG is soldered in, you will have to cut the pin using a sharp, pointed knife. Be careful not to damage the adjacent pins. If you use a socket, bend the pin of the socket and solder your wire to the bent socket pin. Solder all the wires to the perf board according to the schematic diagram. Solder all the wires to their respective pins on the PIA.

All the pins except A3 solder to the PIA. There are several places to get A3. It is pin #22 on the cartridge connector, pin #12 on the CPU (MC6809E), pin #5 on either ROMs or pin #19 on the SAM (MC6883). This brings us to the PIA.

Which PIA should you use? Well, that all depends on whether you want the screen to go back to the inverse mode when you hit Reset. If you put the circuit on the DAC PIA, the software switch will stay in the same position until you turn off the computer or change it. If you put the circuit on the keyboard PIA, the screen will default to the inverse mode every time you hit the Reset key. The choice is up to you.

The poke will be the same, though the addresses will be different depending on which PIA you use. If you have it on the DAC PIA the address to POKE into is $FF30 (65328) and the keyboard PIA has the address set to $FF10 (65296). From now on, if you want to change the state of your screen from one to the other, and the circuit is connected to the DAC PIA, all you have to do is:

```
POKE 65328,0 for Inverted Video
POKE 65328,1 for Normal Video
```

for the keyboard PIA, the command is:

```
POKE 65296,0 for Inverted Video
POKE 65296,1 for Normal Video
```

This modification should not interfere with the normal operation of the computer. It will only invert text in the text mode. None of the graphics mode will be affected and none of the colors of the PSET or PRESET will be changed. One more thing! The chips for this project are not available from Radio Shack, but most electronics parts mail order houses have them. There is nothing special about these chips. If you can't find them, try JDR Microdevices, 1224 S. Bascom Avenue, San Jose, CA 95128. The telephone number is (408) 995-5430.

In closing, I would like to say I have been getting a few letters from readers who are interested in making a computer storage scope using the CoCo. The average price for such a scope starts at about $10,000. The hardware that goes into one also costs big bucks.

To try to make a project of one in this column is not quite possible. First of all, the CoCo is not a fast enough computer to make it worthwhile. Second, the hardware required would run up a bill I don't have the means of paying. And last of all, the time it would take to develop a schematic, you would be lucky to get it by 1999. But keep your ideas coming.

Parts List

Part	Description
IC-1	74LS86
IC-2	74LS151
C1	100pf 50V
Miscellaneous	Perf board, Wire, 14-pin socket, 16-pin socket

ML UTILITY

16K ECB

Machine Code Loder

By Dennis H. Weide

Machine language programs add many features to the 80C that can only be found on more expensive machines. The advantage of these ML programs is that you can have them free. Many magazines print programs in machine language you might want to try.

I have not had the time to sit down and master machine language, assembly language and editor/assemblers, so I wrote a program that allows me to enter an ML listing without an editor/assembler. The program has three sections: one to load the ML listing, one to load a string table and one to print contents of memory.

A Little about ML Listings

First, you will need to understand a little about ML program listings. I am not an expert on machine language and some of my terms may be wrong, but the procedures explained here are valid and have worked for all the listings I have tried. It isn't very hard once you have been through a few lines.

As an example, let's look at Richard Smrcina's *Error Message* program for non-disk users in the January 1984 edition of RAINBOW (Page 169). This is a neat program for printing error messages on the screen.

I won't go into ML programming, but I will tell you a little about how the program listing is laid out. While not all listings follow this format, most of them do and can be loaded using the method described here.

Let's Look at an ML Listing

If you look at the ML listing for error messages, you will notice it contains three major sections. The first section consists of lines 10 to 320; this is the program listing. The four numbers in the left column are the addresses you will be loading. The next two to eight numbers are the opcode and data that will be loaded.

The second section of the program, lines 340 to 850, is the string table. You can tell this is a string table by the heading at Line 330 and the FCC mnemonic for Form Constant Byte or simply a string. The error messages are the strings in the table and they are listed on the right side of the listing. Notice the line following each error message is labeled as FCB or FDB. This marks the end of this string.

The third section of the listing is also a table, however, this is an address table. This section will be loaded the same as the first section. Lines 1130 and 1140 are a string so you must use the string table section of *Codelode* to load these two lines.

The column of numbers just left of center starting with 0010, 0020, etc., is the program line numbers and used for the listing only. Unlike BASIC, ML programs do not need line numbers. They are used only for the programmer's convenience. This column of numbers will be ignored when entering the ML program.

Each address will store a two-character hexadecimal number. Notice that most of the Hex numbers in the data column are longer than two characters. These numbers represent the data in several consecutive addresses. To make things a little clearer, let's look at Line 0060.

```
Line 0060     7DF8 30 8D 0004
```

The actual value at address 7DF8 is 30. For a better explanation, look at the following chart.

Address	Data
7DF8	30
7DF9	8D
7DFA	00
7DFB	04

As you can see, the data listed in the data column is actually the data in four consecutive addresses. The rest of the columns contain programmer comments similar in function to BASIC REMs and the ML opcodes which are similar to the tokens for BASIC statements. You only need to know the information contained in columns up to the line numbers. See how easy it's going to be?

How the Code Loder Works

When you enter data, *Codelode* calculates the number of addresses to be loaded and breaks the data down so it can be loaded into the proper address. It's as simple as that. You do not need to do anything but enter the address you wish to load and the data following it; the BASIC program does the rest.

Now Give It a Try

I have included a sample run of *Codelode* using Mr. Smrcina's *Error Message* program mentioned earlier. You can refer to it as I discuss how to use *Codelode*.

First LOAD and RUN the *Codelode* program. When the menu is displayed, enter '1' to start loading the assembly language listing. You will be asked for the offset value. This is the amount of offset from the address listed in the program. Enter this number in decimal.

Normally, you will load the ML program as it is listed, but if you want to relocate it to another address, enter the difference between the address listed in the program listing and the start address where you want to relocate it. In my sample run, I entered zero so the program will be loaded at the address listed.

If the new address is lower than the start address of the program listing, enter the offset as a negative number. If you wish to enter it as a hexadecimal number, precede the number with &H (Color BASIC's symbol for hexadecimal numbers). Color BASIC users can try this:

```
POKE 25,30:POKE26,1:NEW
```

to clear four graphics pages the same as Extended Color BASIC. Many of the ML programs available are written to fit in these four pages.

After entering the offset address, enter the first address you wish to load. In the program we have chosen as an example, you would type 7df8 and ENTER. The program accepts the addresses and data in a hexadecimal format without &H.

After entering the address, you are ready to input the data. In our example, you would type 308D0004 and ENTER. You will load the first four consecutive addresses with the data entered. The program will respond with the address you entered in decimal format for reference, read each address that was loaded and display it along with its contents. If for any reason you make an error, reenter the bad address and data or the entire line.

At Line 320, you would enter 7E2F for the address and 39 for the data. Since the next section is the start of the table, enter 'M' for the next address. This will return to the main menu.

To enter this table (remember, it is a string table) enter '2' when the menu appears. The first address of this table is 7E30 so enter this when requested, then enter the data (i.e., NEXT WITHOUT FOR) as a string. You must then enter the code following the string. For this string, enter 0D00. The BASIC program will load this value into the proper address and prompt you for the next string address and string to load. Continue this until you have loaded the last address in this table, then enter 'M' for the next address to return to the program menu.

The second table starts at address 7FC5. This table will be entered the same as the first part of the listing except for the last entry. Since this is a string, you must use the string table section of *Codelode*. After loading address 7FF7, enter 'M' for the address to return to the main menu, then enter '2' and load the last string.

Load a Little Faster

You will soon see that you can load much faster if you don't press ENTER at the end of the data to be loaded. You can continue down the data column until you have typed up to 255 characters if you wish. Only enter the address to be loaded when asked to do so — a program like this one can be loaded in 15 minutes or less.

3100	99
5400	85
7305	17
END	54

The listing: CODELODE

```
1000 '
1100 '     MACHINE CODE LOADER
1200 '     BY DENNIS WEIDE (C)198
2
1300 '
1400 CLS
1500 C$="&H"
1600 PRINTSTRING$(32,"=");
1700 PRINTTAB(10)"PROGRAM MENU"
1800 PRINTSTRING$(32,"=")
1900 PRINTTAB(5)"1. ENTER LISTIN
G"
2000 PRINTTAB(5)"2. ENTER STRING
 TABLE"
2100 PRINTTAB(5)"3. PRINT MEMORY
"
2200 PRINTTAB(5)"4. END PROGRAM"
2300 PRINT
2400 PRINTTAB(3)"ENTER ONE OF TH
E ABOVE";
2500 INPUT OA
2600 ON OA GOTO 2700,5600,7700,9
400
2700 CLS
2800 LINE INPUT"ENTER OFFSET VAL
UE (DEC) ";OV$
2900 PRINT
3000 OV=VAL(OV$)
3100 LINE INPUT"ENTER ADDRESS (H
EX) ";A$
3200 IF A$="M" THEN 1400
3300 PRINT
3400 LINE INPUT"ENTER DATA (HEX)
 ";B$
3500 PRINT
3600 A=VAL(C$+A$)
3700 PRINTTAB(4)"DECIMAL ADDRESS
 "A+OV:PRINT
3800 PRINTTAB(11)"WRITE DATA"
3900 FOR X=1 TO LEN(B$) STEP2
4000 D$="0"+(C$+MID$(B$,X,2))
4100 PRINTTAB(5)"ADDR  "HEX$(A+O
V);
4200 PRINTTAB(19)"DATA  "RIGHT$(
D$,2)
4300 POKE A+OV,VAL(C$+MID$(B$,X,
2))
4400 A=A+1
4500 NEXT X
4600 PRINT
4700 PRINTTAB(12)"READ DATA"
4800 A=VAL(C$+A$)
4900 FOR X=1 TO LEN(B$)/2
5000 NV$="0"+HEX$(PEEK(A+OV))
5100 PRINTTAB(5)"ADDR  "HEX$(A+O
V);
5200 PRINTTAB(19)"DATA  "RIGHT$(
NV$,2)
5300 A=A+1
5400 NEXT X:PRINT
5500 GOTO 3100
5600 CLS
5700 LINE INPUT"ENTER ADDRESS (H
EX) ";A$
5800 PRINT
5900 IF A$="M" THEN 1400
6000 PRINT"ENTER TABLE DATA AS S
TRING"
6100 LINE INPUT B$
6200 A=VAL(C$+A$)
6300 Y=0
6400 FOR X=A TO A+LEN(B$)-1
6500 Y=Y+1
6600 POKE X,ASC(MID$(B$,Y,1))
6700 NEXT X
6800 PRINT
6900 LINE INPUT"ENTER RETURN COD
E (HEX) ";B$
7000 Z=1
7100 Z=Z+2
7200 FOR Y=X TO X+LEN(B$)/2-1
7300 POKE Y+OV,VAL(C$+MID$(B$,Z,
2))
7305 Z=Z+2
7400 NEXT Y
7500 PRINT
7600 GOTO 5700
7700 CLS
7800 LINE INPUT"ENTER START ADDR
ESS (HEX) ";A$
7900 PRINT
8000 LINE INPUT"ENTER END ADDRES
S (HEX) ";B$
8100 PRINT
8200 INPUT"DO YOU WANT A HARDCOP
Y (Y/N)";H$
8300 A=VAL(C$+A$)
8400 B=VAL(C$+B$)
8500 FOR X=A TO B
8600 NV$="0"+HEX$(PEEK(X))
8700 IF H$="Y" THEN DV=-2 ELSE D
V=0
8800 PRINT#DV,TAB(5)"ADDR  "HEX$
(X);
8900 PRINT#DV,TAB(19)"DATA  "RIG
HT$(NV$,2)
9000 NEXT X
9100 PRINT
9200 LINE INPUT"   PRESS <ENTER>
 TO CONTINUE";DA$
9300 GOTO 1400
9400 END
```

SAMPLE RUN

```
RUN

==============================
        PROGRAM MENU
==============================

      1. ENTER LISTING
      2. ENTER STRING TABLE
      3. PRINT MEMORY
      4. END PROGRAM

   ENTER ONE OF THE ABOVE? 1

ENTER OFFSET VALUE (DEC) Ø

ENTER ADDRESS (HEX) 7DF8

ENTER DATA (HEX) 3Ø8DØØØ4

    DECIMAL ADDRESS   32248

            WRITE DATA
     ADDR  7DF8     DATA 3Ø
     ADDR  7DF9     DATA 8D
     ADDR  7DFA     DATA ØØ
     ADDR  7DFB     DATA Ø4

             READ DATA
     ADDR  7DF8     DATA 3Ø
     ADDR  7DF9     DATA 8D
     ADDR  7DFA     DATA ØØ
     ADDR  7DFB     DATA Ø4

ENTER ADDRESS (HEX) 7DFC

ENTER DATA (HEX) BFØ192

    DECIMAL ADDRESS   32252

            WRITE DATA
     ADDR  7DFC     DATA  BF
     ADDR  7DFD     DATA  Ø1
     ADDR  7DFE     DATA  92

             READ DATA
     ADDR  7DFC     DATA  BF
     ADDR  7DFD     DATA  Ø1
     ADDR  7DFE     DATA  92

ENTER ADDRESS (HEX) 7DFF

ENTER DATA (HEX) 39

    DECIMAL ADDRESS   32255

            WRITE DATA
     ADDR  7DFF     DATA  39

             READ DATA
     ADDR  7DFF     DATA  39

ENTER ADDRESS (HEX) 7EØØ

ENTER DATA (HEX) BDA7E9

    DECIMAL ADDRESS   32256

            WRITE DATA
     ADDR  7EØØ     DATA  BD
     ADDR  7EØ1     DATA  A7
     ADDR  7EØ2     DATA  E9

             READ DATA
     ADDR  7EØØ     DATA  BD
     ADDR  7EØ1     DATA  A7
     ADDR  7EØ2     DATA  E9

ENTER ADDRESS (HEX) M

==============================
        PROGRAM MENU
==============================

      1. ENTER LISTING
      2. ENTER STRING TABLE
      3. PRINT MEMORY
      4. END PROGRAM

   ENTER ONE OF THE ABOVE? 2

ENTER ADDRESS (HEX) 7E3Ø

ENTER TABLE DATA AS STRING
NEXT WITHOUT FOR

ENTER RETURN CODE (HEX) ØDØØ

ENTER ADDRESS (HEX) 7E42

ENTER TABLE DATA AS STRING
SYNTAX

ENTER RETURN CODE (HEX) ØØ

ENTER ADDRESS (HEX) 7E49

ENTER TABLE DATA AS STRING
RETURN WITHOUT GOSUB

ENTER RETURN CODE (HEX) ØDØØ

ENTER ADDRESS (HEX) 7E5F

ENTER TABLE DATA AS STRING
OUT OF DATA

ENTER RETURN CODE (HEX) ØØ

ENTER ADDRESS (HEX) M

==============================
        PROGRAM MENU
==============================

      1. ENTER LISTING
      2. ENTER STRING TABLE
      3. PRINT MEMORY
      4. END PROGRAM

   ENTER ONE OF THE ABOVE? 4
OK
```

BYTE MASTER

16K
ECB

Flashing In 6809

By R. Bartly Betts

With Programs By Chris Bone

If you use *Bytescreen*, you may have discovered there is a way to crash the program. It isn't of major concern because it can only happen if you try to change the PMODE value when the cursor is located on the bottom half of the screen. It most likely results from *Bytescreen* trying to scroll when you are not in PMODE 4, and there just isn't anything to scroll.

You also may have discovered you can get extra wide and extra high characters in PMODE 2. You do need to be careful regarding cursor position before changing the PMODE, however.

Bytescreen Upgrade

I have had a lot of interesting correspondence and phone calls from readers lately, and I do appreciate hearing from you. I was especially glad to hear from George J.P. Dabbene of Ottawa, Ontario, who submitted some nifty upgrades for *Bytescreen*. I expect good things to come out of Canada: first, that is where I came from, and second, Canadians should have more time for programming. (If you had a choice of writing computer programs or shoveling the walk in 40 below zero weather, which would you do?) I received George's program back in February, so I know what he was doing during the long winter months. Listing 4, at the end of this column, shows George's upgrades and tells what they do.

The Ground Work

Before getting into upgrades, you need to spend some more time on the basics. As promised last month, I am going to provide a little more programming meat on which to sharpen your teeth. Beginning this month, the next few columns will deal with doing things with the text screen, the keyboard and interfacing machine language programs to BASIC.

You've probably noticed I have dropped the machine language arithmetic series I began. It seems I was getting over the head of many readers and so decided to detour for a while.

This month's program lets you flash up to 255 characters on the text screen for a selected number of times. I have also written two sample BASIC programs that show how the process works. To get things started, power-up your computer and your editor/assembler, then enter the following code:

Listing 1: FLASH
(A program to reverse a selected portion of the text screen a selected number of times.)

```
                    ØØØØ1 ***********************
                    ØØØØ2 *     FLASH SCREEN    *
                    ØØØØ3 * BY R. BARTLY BETTS  *
                    ØØØØ4 ***********************
                    ØØØØ5 *
                    ØØØØ6 *
ØEØØ                ØØØ1Ø               ORG     $EØØ
ØEØØ                ØØØ2Ø STOR1         RMB     2
ØEØ2                ØØØ3Ø STOR2         RMB     2
ØEØ4                ØØØ4Ø STOR3         RMB     1
ØEØ5 4F             ØØØ5Ø START         CLRA
ØEØ6 F6   ØEØØ      ØØØ6Ø               LDB     STOR1
ØEØ9 C3   Ø4ØØ      ØØØ7Ø               ADDD    #1Ø24
ØEØC 1F   Ø1        ØØØ8Ø               TFR     D,X
ØEØE F6   ØEØ2      ØØØ9Ø               LDB     STOR2
ØE11 A6   84        ØØ1ØØ LOOP          LDA     ,X
ØE13 88   4Ø        ØØ11Ø               EORA    #64
ØE15 A7   8Ø        ØØ12Ø               STA     ,X+
ØE17 5A             ØØ13Ø               DECB
ØE18 26   F7        ØØ14Ø               BNE     LOOP
ØE1A 7A   ØEØ4      ØØ15Ø               DEC     STOR3
ØE1D 27   ØA        ØØ16Ø               BEQ     DONE
ØE1F 31   31        ØØ17Ø TIME          LEAY    -$F,Y
ØE21 1Ø8C ØØØF      ØØ18Ø               CMPY    #$F
ØE25 25   DE        ØØ19Ø               BLO     START
ØE27 2Ø   F6        ØØ2ØØ               BRA     TIME
ØE29 39             ØØ21Ø DONE          RTS
          ØØØØ      ØØ22Ø               END
ØØØØØ TOTAL ERRORS
```

BASIC Interfacing — Step 1

This program is designed to show one way of interfacing to BASIC programs. It lets you store three values in memory: the beginning screen address of the characters you wish to flash, the number of characters you wish to flash, and the number of times they flash. The values must be provided from BASIC to direct the machine language program.

For instance, if you wish to flash characters beginning at PRINT location 101, POKE 101 into memory location $E00. If you wish 10 characters, or screen positions, to flash, POKE 10 in memory location $E02. If you wish the characters to flash 15 times, POKE 15 into memory location $E04. The three RMB statements in lines 20, 30 and 40 reserve memory for these values. That is the purpose of RMB instructions, to reserve a specified number of bytes for data.

The location of the RMB instructions in a program determines where in memory the storage bytes will be reserved. In this program, the storage bytes are the first bytes. No matter where you place the machine language code, these storage bytes are always at the beginning of the program.

Although the code is relocatable, I have placed it in the graphics screen memory to make it compatible with both 16 and 32K computers. You can select another location if you wish, for example, the top of memory. If you do so, use the CLEAR statement to reserve sufficient memory before using *Screen Flash* with a BASIC program.

The Lowdown on *Screen Flash*

Line 30 loads the first storage byte into Register B (this value must have been poked into $E00 from BASIC). A value of 1024 is then added to Register D, which comprises both registers A and B, and the result is transferred to Register X.

Adding 1024 to the value previously loaded into 'B' creates a pointer to the actual screen memory where you wish the flash process to take place. It is this pointer that is stored in Register X, where it is increased by one byte as each character is inverted.

Next, Register B is loaded with the number of characters, or screen locations, you wish to flash. This value must also be POKEd into the second RMB storage area from BASIC. Lines 100 through 140 load each screen character value into Register A, perform an exclusive OR operation with 64 to reverse the character, and load it back into the screen memory from where it came. After each invert operation, the value in 'B' is decreased by one. If 'B' is not equal to zero, Line 140 causes the process to repeat.

When 'B' does equal zero (all of the designated characters having been inverted) the value in Storage Area 3 is decreased. If that value is now zero, the process ends. If it is not zero, the program runs through a delay loop beginning at Line 170. Without the delay loop the time between character inversions is too short to notice. After the delay, the process branches back to START to begin the whole operation again.

Taking Notes

There are a few points about the instructions used in this program that are noteworthy:

1) The following information tells you how the exclusive OR function in Line 110 works. To demonstrate, I have selected the uppercase character 'B' which has an ASCII code value of 66 (in binary, this is 1000010). An exclusive OR of the character 'B' with 64 compares the bits of 66 and 64. When one, but not the other of the two bits being compared is set (equals one), the result bit is set (equals one). Thus, an exclusive OR of 66 with 64 gives the result:

```
01000010      — 66 decimal
01000000      — 64 decimal
————————
00000010      — exclusive OR result = 2 decimal
```

As you see, the result is '2', which produces a lowercase or inverse 'B' when stored in screen memory. If you wish to test this from BASIC type POKE 1024,2. When the same screen location is again subjected to an exclusive OR, the result is:

```
00000010      —  2 decimal
01000000      — 64 decimal
————————
01000010      — exclusive OR result = 66 decimal
```

The original value is restored. This same process holds true for all text screen characters and provides the way to toggle between upper- and lowercase, or regular and inverse, characters. Doing this in machine language is fast enough to make all of the specified characters appear to flash at the same time, even if you set the maximum 255 characters.

2) Notice that it is not necessary to compare Register B to zero before implementing the BNE (Branch not Equal), BEQ (Branch Equal) or the BLO (Branch on Lower) instructions, lines 140, 160 and 190, respectively. You often see programs that include a CMPB #0 instruction before branches, but the comparison instruction is only necessary if you are comparing to a number other than zero, as in Line 180.

3) The transfer instruction TFR only transfers data between equally sized registers, 8-bit to 8-bit, or 16-bit to 16-bit.

4) You can use the decrement instruction DEC to decrease values in memory locations as well as in registers, as in Line 150. DEC always decreases in steps of one. The instruction to increment, INC, always increases in steps of one.

5) The LEA instruction (Load Effective Address) used in Line 170 acts as a delay loop. By giving an offset of -$F, or decimal -15, the value in the 'Y' register is decreased by 15 each time through the loop. Changing this value changes the length of the delay between flashes.

Connecting with BASIC

The following BASIC listing shows how the assembly language program might be used. It creates a demonstration menu and uses *Flash Screen* to highlight one of the menu selections by flashing the characters 15 times.

Listing 2: DEMO 1

```
10 DEF USR0=&HE05
30 CLS
40 PRINT TAB(10)"THE MENU"
50 PRINT
60 FOR T=1 TO 4
70 READ A$
80 PRINT TAB(5)A$
90 NEXT
100 POKE &HE00,101
110 POKE &HE02,17
120 POKE &HE04,15
150 SOUND 200,2
160 A=USR(0)
170 POKE &HE04,1
200 DATA 1. ADDITION,2. SUBTRACT
ION,3. MULTIPLICATION, 4. DIVISI
ON
```

Notice the execution address for the *Flash Scree* program is &E05. This is because of the RMB instructio at the beginning of the program. If you attempt to execut the program at $E00, it crashes. The entire interfacin between this BASIC program and the *Flash Screen* machin language program is in lines 10, 100-120 and Line 60. Lin 10 defines the execution address, lines 100-120 POKE th desired values into the reserved memory bytes and Lin 60 calls *Flash Screen*.

The following BASIC listing shows how to go a step furthe with BASIC interfacing. This program sets the number o flashes to two and uses a BASIC loop to repeatedly flas the desired text. The BASIC loop incorporates a soun routine in the flashing process.

Listing 3: DEMO 2

```
10 DEF USR0=&HE05
20 POKE &HE04,TM
30 CLS
40 PRINT TAB(10)"THE MENU"
```

```
50 PRINT
60 FOR T=1 TO 4
70 READ A$
80 PRINT TAB(5)A$
90 NEXT
92 P=69
95 FOR Z=1 TO 4
100 POKE &HE00,P
110 POKE &HE02,17
120 POKE &HE04,1
140 FOR T=1 TO 8
150 SOUND 200,2
160 A=USR(0)
170 POKE &HE04,1
180 NEXT T
190 P=P+32:NEXT Z
200 DATA 1. ADDITION,2. SUBTRACT
ION,3. MULTIPLICATION, 4. DIVISI
ON
```

The additional POKE in Line 170 is to ensure that once a portion of the screen is flashed, it doesn't remain inverted.

Forging Forward

The *Flash Screen* program was purposely left incomplete. For instance, it only flashes characters on the top half of the text screen; you cannot POKE a value greater than 255 into the STOR1 byte.

Note, however, the STOR1 RMB sets aside two bytes for storage. This provides a way of accessing the entire text screen with the program. Do you know how to do it? Give it a try. I would be pleased to see your results.

How about a feature to flash a particular sequence of characters at any location on the screen, such as an arrow (←)? Come up with some ideas of your own and experiment with the *Flash Screen* program.

Bytescreen and Underline

Now for the promised revisions to *Bytescreen* (Listing 4). Mr. Dabbene has supplied some additional code that provides an underline feature and some new characters. Note that revised program lines are included in parentheses. Do not type the parentheses when changing the lines in your source listing of *Bytescreen*.

New lines to the program are numbered to fit between the existing lines. Be careful to type the correct line number. Use the normal 'I' (for INSERT) and 'E' (for EDIT) commands to make the changes. When finished, check them carefully, reassemble the listing and you are ready to use your new, advanced *Bytescreen*.

You can now produce an underline character by pressing the SHIFT and up-arrow keys together. You can use an underline in a screen print routine by using CHR$(31). For instance, to underline the word "never" in "I told you to never touch my computer," type:

```
PRINT "I TOLD YOU TO" CHR$(31) "NEVER" CHR$(31)
"TOUCH MY COMPUTER"
```

CHR$(31) turns the underline on and is then used again to turn the underline off.

The new code also contains a routine which only recognizes valid charactes. Bytescreen previously printed garbage if you attempted to print codes for which no character existed. Now it ignores the code. Also, several new characters have been added:

ASCII Code	Character
123	left brace
124	right brace
125	pi symbol
126-128	graphics characters

Adding new characters to *Bytescreen* is easy and you may wish to try some yourself. If I can find the room in the future, I will try to provide information on how to do so. As it is, you can probably discover how to do so on your own.

I don't know what I can do for those of you who do not have editor/assemblers. I do not have an assembled listing that provides the machine language code. Even if you have the code, you must reenter the entire program with the new code inserted in the proper places. This is just one time

Listing 4: BYTESCRN
(Changes to the *Bytescreen* source code)

Bytescreen alterations lines which already exist, but need changing, are enclosed in parentheses. Do not type these parentheses.

```
(2Ø              ORG      $7C67  *ADJUSTMENT TO ACCOMMODATE NEW CODE)

(27Ø PRETX       PULS     D,X    *NEW LABEL)

(44Ø             LBSR     LETT2  *LONG BRANCH-LABEL CHANGE)

711              CMPA     #31    *CODE FOR UNDERLINE
712              BNE      NOTUL  *IGNORE
713              LDA      #$FØ   *(SET) SWITCH ON VALUE
714              CMPA     ULFLAG
715              BNE      ULSET
716              BSR      ULCLR  *(CLR) UNDERLINE SWITCH OFF
717              BRA      PRETX  *CLEAR STACK/RETURN TO PARSE
718  ULSET       BSR      CTRL   *(SET) UNDERLINE SWITCH ON
719              BRA      PRETX  *CLEAR STACK/RETURN TO PARSE
(72Ø NOTUL       CMPA     #8     *NEW LABEL)

(76Ø             LBSR     LETT2  *LONG BRANCH-NEW LABEL)

(78Ø             LBSR     LETT2  *LONG BRANCH-NEW LABEL)

(1Ø5Ø            BSR      LETT2  *LABEL CHANGE)

1Ø71 ULCLR       LDA      #$ØF   *(CLR) SWITCH OFF VALUE
1Ø72 CTRL        STA      ULFLAG
1Ø73             LDB      #94    *LENGTH OF NON-GRAPHIC CHAR SET
1Ø74             LDX      #TABLE *START OF TABLE
1Ø75 CTRLOP      LEAX     3,X    *SKIP 1ST 3 BYTES
1Ø76             LDA      ULFLAG *GET SWITCH VALUE
1Ø77             CMPA     #$ØF   *? OFF
1Ø78             BNE      SET    *NO, SET CHAR ON
```

```
1Ø79            ORA     ,X      *MAKE LSB=$F (OFF)
1Ø8Ø            BRA     PROCES
1Ø81 SET        ANDA    ,X      *MAKE LSB=Ø
1Ø82 PROCES     STA     ,X+     *REPLACE BYTE AFTER ADJUST
1Ø83            DECB            *AND POINT TO NEXT BYTE
1Ø84            BNE     CTRLOP  *DO ALL CHARS IN SET
1Ø85            RTS
(1Ø86 RETURN    JMP     $1ØØØ   *WAS AT LINE 1Ø8Ø)

(11ØØ           BNE     LETT1   *CHARACTER SET LIMITS)
11Ø5            BSR     ULCLR   *CAR. RET. CLS UNDERLINE

(112Ø           BSR     LETT2)

1171  LETT1     CMPA    #32     *LOWEST VALID CHARACTER
1172            LBLO    PRET    *IGNORE
1173            CMPA    #128    *HIGHEST VALID CHARACTER
1174            LBHI    PRET    *IGNORE

(239Ø           FDB     $FFED   *CHR$(48) SLASH)
(24ØØ           FDB     $BFFF)

(329Ø           FDB     $FFFB   *CHR$(92) BACKSLASH)
(33ØØ           FDB     $DEFF)

(333Ø           FDB     $FD8D   *CHR$(94) CARET)
(334Ø           FDB     $DDFF)

(343Ø           FDB     $FFF1   *CHR$(99) c)
(344Ø           FDB     $771F)

(351Ø           FDB     $FFØ6   *CHR$(1Ø3) g)
(352Ø           FDB     $ØE8F)

(369Ø           FDB     $FFØ6   *CHR$(112) p)
(37ØØ           FDB     $Ø77F)
(371Ø           FDB     $FF86   *CHR$(113) q)
(372Ø           FDB     $AEEF)

(387Ø           FDB     $FF66   *CHR$(121) y)
(388Ø           FDB     $9D9F)
39Ø1            FDB     $DBB3   *CHR$(123) LEFT BRACE
39Ø2            FDB     $BBDF
39Ø3            FDB     $BDDC   *CHR$(124) RIGHT BRACE
39Ø4            FDB     $DDBF
39Ø5            FDB     $FFØA   *CHR$(125) PI SYMBOL
39Ø6            FDB     $AAAF
39Ø7            FDB     $AAAA   *CHR$(126) GRAPHIC
39Ø8            FDB     $AAAA
39Ø9            FDB     $5555   *CHR$(127) GRAPHIC
391Ø            FDB     $5555
3911            FDB     $Ø      *CHR$(128) GRAPHIC
3912            FDB     $Ø

3915 MASK       RMB     2       *MOVED FROM LINE 391Ø
```

when an editor/assembler is needed.

I can make the same offer as before. If you wish to send $10, I will mail you a tape or disk version of the program. For those of you who have already sent $10 for the program and want updates as they come out, I will supply them at no cost if you send a diskette or tape and the return postage. If you do not want to keep sending repeatedly, wait until all of the upgrades have been made.

Farewell to Chris

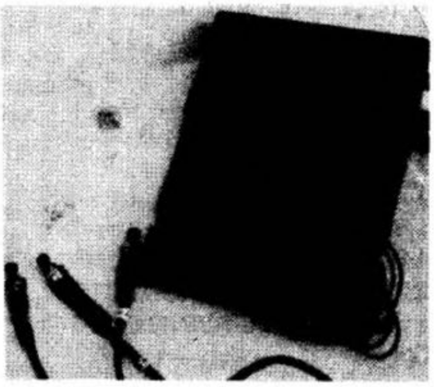
Although Chris plans to provide a couple more upgrades to "Byte Master," he will soon be leaving the column. He is getting too busy and is returning to Canada this summer. Chris' schooling and the distance make it impossible for him to continue with us. I will miss him but feel we can still keep battering away at assembly language. He has certainly helped me get started and finds a lot of my errors. I will have to work harder without him.

You can write to me by addressing your letters to:
R. Bartly Betts
2251 Lipscomb
Fort Worth, Texas 76110
Phone (817) 924-3725

(If you wish a reply, please include a self-addressed, stamped envelope. Please do not call collect or expect me to return calls unless I can call collect.)

THE COCOCONNECTION

by John Poxon & Graham Morphett



Well at last the CoCoConnection has become a reality, and we are able to both show you what it looks like and present our first program to support it.

This unit allows you to connect most electrical or mechanical items to CoCo, and will accept data from external sources as well as switch external units.

The best way to talk about the many uses for this unit is to describe in detail projects we are undertaking with it.

The first of these is a "Quizmaster".

You could fit 32 buttons to the CoCoConnection, but for our demonstration unit, we fitted eight.

These are arranged as per the photograph, in accordance with the instructions set out in the fairly comprehensive instructions.

The wires have a normally open press button soldered on at one end. We numbered each wire, but you could have a routine in the program which would ask each button holder to identify themselves, thereby negating the need to go through the numbering thing at the beginning of the program.

The following program is loaded and RUN. It asks a question and the first one to press his/her button is asked by the computer to enter their answer.

At demonstrations, we elect a typist to do this, as all those wires get tangled very easily!

At the end of the questions, a report is presented on the screen, which also indicates the winner/s. This part of the program is quite useful in its own right, and deserves study for inclusion in other programs.

Keep in mind the demonstrative nature of this program when looking at the questions we run - you can change them very easily by retyping each data line from 810 onwards. The format is LINE NUMBER DATA QUESTION,ANSWER.

BASIC is really a bit slow for this sort of job, so in due course, we will have a FORTH program to do this job too.

THE LISTING:

```
1 '*COCOCONNECTION DEMONSTRATION
  *************GAME************
  *JOHN POXON & GRAHAM MORPHETT
  *******   26/8/85   *********
2 GOTO10
3 SAVE"CONDEMO:2":DIR2:STOP
4 PLAY"01;L255;ABCDE;L255.;FGABC
DEFG":GOTO4
5 R=RND(7):IFR=1THENR$="A" ELSE
IF R=2 THENR$="B" ELSE IF R=3THE
N R$="C" ELSE IF R=4 THEN R$="D"
 ELSE IF R=5THEN R$="E" ELSE IF
R=6THEN R$="F" ELSE R$="G"
6 A$=A$+R$:N=N+1:IFN=NN THEN7 EL
SE 5
7 L=RND(255):IFL<200THEN7 ELSE P
LAY"01;L255":PLAY A$
8 A$="":N=0:NN=RND(10):GOTO5
10 CLS4:POKE65495,0
20 CLEAR2000:DIMZ$(8),ZZ(8)
40 PRINT@104,"****************";
50 PRINT@136,"***SWITCHGAME***";
60 PRINT@168,"***    BY    ***";
70 PRINT@200,"***JOHN POXON***";
80 PRINT@232,"****************";
90 FORT=0TO70:PLAY"01;L255;ABCDE
FG":NEXTT
100 '***BEGIN***
130 CLS
135 PRINT@64," THIS IS A SMALL D
EMONSTRATION":PRINT"   OF THE AB
ILITIES OF THE":PRINTTAB(8)"COCO
CONNECTION.":PRINTTAB(9)"IF IT I
S NOT":PRINT"    INSERTED INTO T
HE COLOUR          COMPUTER, SWITC
H OFF THE          COMPUTER BEFORE
 INSERTING.":GOSUB185
136 PRINT"ENSURE ALL WIRES ARE F
IRMLY      CONNECTED AND ARE NOT
LIKELY TO PULL THE COCOCONNECTIO
N FROM THECOMPUTER."
139 GOSUB185
140 GOSUB200
150 GOSUB300
160 GOTO400
180 END
185 PRINT@448,"PRESS <ENTER> TO
CONTINUE";:PLAY"01;L255;ABCDEFGA
BCDEFG":I$=INKEY$:IFI$=""THEN185
ELSE CLS:RETURN
190 '
200 '***INITIALISATION***
220 POKE 65409,0:POKE 65408,0:PO
KE 65409,60
230 POKE 65415,0:POKE 65414,0:PO
KE 65415,60
240 RETURN
290 '
300 '***STORE STUDENT NAMES***
310 'IT WOULD BE HANDY TO BE ABL
E TO LOAD IN NAMES FROM A STUDEN
T FILE HERE TO NEGATE THE NEED T
O INPUT THE SAME NAMES EACH TIME
!
330 CLS:PRINT@69,"STUDENT NAME I
NPUT."
340 PRINT
350 FORX=1TO8
360 PRINT"STUDENT #";X;:LINEINPU
TZ$(X)
390 NEXTX
395 RETURN
399 '
400 '***ASK QUESTIONS***
410 READQ$,AN$
420 IFQ$="Q"THEN700
430 CLS:PRINTQ$
450 A=PEEK(65408)
460 B=255-A
465 PLAY"01;L255;ABCDEFG"
470 IF B=0 THEN 450
480 GOSUB500:GOTO410
490 GOTO450
495 '
500 '**FIND FIRST BUTTON PUSHING
                 STUDENT
550 IFB=1THENX=1:GOSUB650
560 IFB=2THENX=2:GOSUB650
570 IFB=4THENX=3:GOSUB650
580 IFB=8THENX=4:GOSUB650
590 IFB=16THENX±5:GOSUB650
600 IFB=32THENX=6:GOSUB650
610 IFB=64THENX=7:GOSUB650
620 IFB=128THENX=8:GOSUB650
630 RETURN
640 '
650 '***PRINT NAME**
660 CLS:PRINT@96,Z$(X);" YOU WER
E FIRST":PRINTQ$:LINEINPUTI$
670 IFI$=AN$THEN PRINT"WELL DONE
 ";Z$(X):FORT=1TO2500:NEXT:ZZ(X)
=ZZ(X)+1:CLS:GOTO695
690 PRINT"I'M SORRY "Z$(X)" EVER
YONE TRY AGAIN!":FORT=1TO2500:NE
XT:CLS:GOTO430
695 IFHS<ZZ(X)THEN HS=ZZ(X)
696 GOTO410
699 '
700 '***FINAL SCORES***
710 CLS:PRINTTAB(6)"YOUR RESULTS
"
715 PRINT"   NAME","SCORE"
720 FORX=1TO8
730 PRINTZ$(X),ZZ(X);:IF ZZ(X)=H
S THEN PRINT"*" ELSE PRINT
740 NEXTX
745 PRINT:PRINTTAB(5)"'*' - WINN
ER!"
750 GOSUB185
760 PRINT"THIS UNIT ALLOWS THE C
OCO TO":PRINT"CONTROL MOST ELECT
RICAL OR MECH-ANICAL ITEMS. THIN
GS LIKE ROBOTSMODEL TRAINS BURGU
LAR ALARMS ANDSCIENCE EXPERIMENT
S ARE SUITABLESUBJECTS FOR THE"
765 PRINTTAB(8)"COCOCONNECTION."
:GOSUB185
770 PRINT"THANK YOU FOR PARTICIP
ATING                   BYE!":PO
KE65494,0:END
799 '
800 '***DATA FOR QUESTIONS***
810 DATA"WHAT IS THE CAPITAL OF
                QUEENSLAND?","BRI
SBANE"
820 DATA"WHAT FRUIT IS COFFS HAR
BOUR                FAMOUS FOR?"
,"BANANAS"
830 DATA"WHICH TOWN IN QUEENSLAN
D HAS THE BRIGHTEST STUDENTS?",W
ARWICK
840 DATA"HOW DO YOU SPELL 'BEER'
?","XXXX"
850 DATA"AT WHICH SHOP IN WARWIC
K CAN YOUBUY AUSTRALIAN RAINBOW
MAGAZINE?","TANDY"
860 DATA WHAT DUST HAS TRADITION
ALLY BEENASSOCIATED WITH TEACHER
S?,CHALK
870 DATA WHY TEACH?,MONEY
880 DATA WHAT IS THE NICK NAME O
F THE    TANDY COLOUR COMPUTER?,
COCO
990 DATA"Q","Q",
```

EZ LOADER

by John Nicolettos

Cassette recorders provide a very effective and easy to use mass storage capability for the COCo. While cassette recorders do not have all the capabilities of a floppy, they do not cost $300 either. A program can be loaded by simply typing CLOAD. The CoCo. will turn the cassette on, load the program, and then turn the cassette off.

All this seems simple until you lose track of where your program starts and you try loading in the middle of a program. CLOADING resuls in I/O error after I/O error. At this stage I usually pull the motor control and the earphone plugs and listen for the start of the program. Typically, I'll reconnect them, type CLOAD, and load the wrong program. Of course, I could have typed MOTOR ON and then AUDIO ON instead. However, by this time my patience has worn thin and pulling the plugs is faster and more satisfying. EZ loader was written to save both me and my recorder from undue wear.

EZ loader will CLOAD any BASIC program either binary or ASCII by simply pressing the down arrow key. The prompt SEARCHING will appear in the upper left hand corner, to let you know that EZ loader is working. If EZ loader was started in the middle of a program the S will alternate between normal and inverted video. The real beauty of EZ loader is that it will not allow an I/O error to occur. Instead it will continue to search until the beginning of the next program. When EZ loader finds the beginning of a program it will replace the SEARCHING prompt with a flashing F and the name of the program. EZ loader will then load the BASIC program. EZ loader will not load machine language programs, nor will it allow you to select the program to be CLOADed.

Machine Language Listing:

```
7EF4               00100        ORG   32500
7EF4 30  8D 002B  00110 START  LEAX  MERGE,PCR        PUT MERGE ADDR INTO
7EF8 BF  0168     00120        STX   $168                  JMP TABLE
7EFB 6F  8D 0003  00130        CLR   SW,PCR           SET SWITCH FOR MERGE
7EFF 39           00140        RTS
7F00              00150 STR    RMB   2                START POINTER STORAGE
7F02              00160 SW     RMB   1                SWITCH STORAGE
7F03     4D       00170 PMG    FCC   /MERGE INITIATED/
7F12     0D       00180        FCB   $0D
7F13     4D       00190 PRT    FCC   /MERGE COMPLETED/
7F22     0D       00200        FCB   $0D
7F23 81  5C       00210 MERGE  CMPA  #$5C             LOOK FOR CONTROL CHR
7F25 26  36       00220        BNE   EXIT             EXIT IF NOT CONTROL
7F27 6D  8D FFD7  00230        TST   SW,PCR           CHECK SWITCH SETTING
7F2B 26  1B       00240        BNE   RST
7F2D 9E  19       00250        LDX   $19              GET START ADDR
7F2F AF  8D FFCD  00260        STX   )STR,PCR         STORE IN STR
7F33 9E  1B       00270        LDX   $1B              GET END ADDR
7F35 30  1E       00280        LEAX  -2,X             OVERWRITE END FLAG
7F37 9F  19       00290        STX   $19
7F39 31  8C C7    00300        LEAY  PMG,PCR
7F3C 63  8D FFC2  00310        COM   SW,PCR           TOGGLE SWITCH
7F40 8D  1E       00320        BSR   PRINT            GO TO SCRN PRINT SUB
7F42 8E  ABEF     00330        LDX   #$ABEF
7F45 7E  AC79     00340        JMP   $AC79            RETURN TO BASIC
7F48 AE  8D FFB4  00350 RST    LDX   )STR,PCR         GET START ADDR
7F4C 9F  19       00360        STX   $19              PUT IN POINTER
7F4E 31  8C C2    00370        LEAY  PRT,PCR          GET PROMPT ADDR
7F51 6F  8D FFAD  00380        CLR   SW,PCR           TOGGLE SWITCH
7F55 8D  09       00390        BSR   PRINT            GO TO SCRN PRINT SUB
7F57 8E  ABEF     00400        LDX   #$ABEF
7F5A 7E  AC79     00410        JMP   $AC79            RETURN TO BASIC
7F5D 7E  8273     00420 EXIT   JMP   $8273            NORMAL RETURN
7F60 C6  10       00430 PRINT  LDB   #$10             # OF CHRS IN PROMPT
7F62 A6  A0       00440 PR1    LDA   ,Y+              GET CHR
7F64 BD  A282     00450        JSR   $A282            PRINT CHR ON SCREEN
7F67 5A           00460        DECB
7F68 26  F8       00470        BNE   PR1
7F6A 39           00480        RTS
         0000     00490        END
00000 TOTAL ERRORS

EXIT    7F5D
MERGE   7F23
PMG     7F03
PR1     7F62
PRINT   7F60
PRT     7F13
RST     7F48
START   7EF4
STR     7F00
SW      7F02
```

PROGRAM DESCRIPTION

Program listing 1. contains the assembly language program for EZ loader. Lines 100-120 replace the normal keyboard jump address with the start address of the program CLOAD (lines 130-380). Each character inputed, by the keyboard, is checked by lines 130 and 140. If the input character is not the down arrow then the program branches to EXIT. EXIT will return control to the COCO by jumping to location hex 8CF1, the normal return from a keyboard input.

When the down arrow has been entered then CLOAD is executed. Lines 150-210 will clear the screen and print the prompt SEARCHING. The CoCo is placed in the proper mode and the cassette load address identified by lines 220-250. A block of cassette data is read in and a test made to see if it contained a file name by lines 260-280. If the block did not contain a file name the process is repeated.

When a file name is found then the screen is cleared, the cursor moved to the right two positions, and the nam printed by lines 290-320. Lines 330-350 prepare the CoCo for CLOADing a BASIC program. Line 360 sends the CoCo to its normal CLOAD ROM routine.

THE BASIC DRIVER

Program listing 2. contains a BASIC language driver for EZ loader. It is a friendly program which will instruct you on the use of EZ loader. Itwill also determine the available memory and poke the machine language program has into the highest 80 bytes. There is an error check on your DATA statements and the program will protect and execute the machine language program.

The instructions for EZ loader are printed by lines 10-160. Line 170 provides a loading prompt for poking the machine language program. The start address for the machine language program is determined by lines 180 and 190. Lines 200-220 poke EZ loader into the highest 80 bytes of available RAM. Line 210 contains a simple arithmetic check on the DATA statements. If an error was made then line 230 will cause the error prompt to be printed. Lines 240 and 250 will execute and protect EZ loader from BASIC. A termination prompt is displayed by line 260 and 270.

OPERATING INSTRUCTIONS

Simply run the BASIC driver, after you have typed and CSAVED it. The instruction pages will appear first. Press ENTER when you have finished reading each page. The machine language program will then be loaded and the prompt LOADING displayed. The end of program prompt will signal the completion of loading.

You may now CLOAD any BASIC program by merely pressing the down arrow. The computer will respond by clearing the screen and printing the SEARCHING prompt. If the S is flashing then you have tried to CLOAD in the middle of a program. You may either let the CoCo continue until the end of the program or you may stop the recorder, rewind it, and try to find the start of the current program. In either case the CoCo will not display an I/O error.

When the CoCo finds the beginning of a program then the prompt will be overwritten with an F and the program name. The F will flash as the program is loaded into memory. Remember that EZ loader will not load machine language programs, only BASIC programs.

The Listing:

```
1 '    **************************
2 '    *        EZ LOADER       *
3 '    *           BY           *
4 '    *   JOHN L. NICOLETTOS   *
5 '    *    FEBRUARY 1, 1984    *
6 '    * ALL  RIGHTS RESERVED   *
7 '    **************************
8 '
9 '
10 CLS
20 PRINTTAB(11)"ez"CHR$(128)"loader"
:PRINT@32,STRING$(32,131);
30 PRINT"   EZ LOADER WILL AUTOMATIC
ALLY":PRINT"LOAD, PROTECT, AND EXECU
TE A"
40 PRINT"MACHINE LANGUAGE CASSETTE":
PRINT"LOADING UTILITY.  TO cload A"
50 PRINT"BASIC PROGRAM SIMPLY PRESS
THE":PRINT"DOWN ARROW KEY.  EZ LOADE
R WILL"
60 PRINT"FIND AND CLOAD THE NEXT BAS
IC  ":PRINT"(BINARY OR ASCII) PROGRA
M."
70 PRINT"   THE PROMPT searching WIL
L":PRINT"APPEAR IN THE UPPER LEFT CO
RNER.";
80 PRINT"WHEN EZ LOADER FINDS AND CL
OADS":PRINT"A PROGRAM THE PROGRAM'S
NAME"
90 PRINT"WILL REPLACE THE PROMPT."
100 PRINT@484,"PRESS ENTER TO CONTIN
UE";:LINEINPUT ZZ$
110 CLS
120 PRINTTAB(11)"ez"CHR$(128)"loader
":PRINT@32,STRING$(32,131);
130 PRINT:PRINT
140 PRINT:PRINT"   EZ LOADER WILL N
OT LOAD"
150 PRINT"   MACHINE LANGUAGE PROGRA
MS."
160 PRINT@484,"PRESS ENTER TO CONTIN
UE";:LINEINPUT ZZ$
170 CLS:PRINT@259,"L   O   A   D   I
   N   G"
180 ED=PEEK(39)*256+PEEK(40)
190 ST=ED - 79
200 FOR X=ST TO ED
210 READ D:POKE X,D:SUM=SUM+D
220 NEXT X
230 IF SUM <> 8823 THEN CLS:PRINT@26
3,"!!!DATA ERROR!!!":END
240 EXEC ST
250 CLEAR 200,ST
260 CLS:PRINT@267,"ez"CHR$(128)"load
er"
270 PRINTTAB(11)"completed"
280 END
290 DATA 48, 141, 0, 4, 191, 1, 107,
 57, 129, 10
300 DATA 38, 56, 189, 169, 40, 48, 1
41, 0, 52, 198
310 DATA 9, 166, 128, 189, 162, 130,
 90, 38, 248, 142
320 DATA 1, 218, 159, 126, 16, 142,
255, 255, 16, 159
330 DATA 104, 189, 167, 1, 218, 124,
 38, 220, 189, 169
340 DATA 40, 12, 137, 12, 137, 189,
166, 156, 15, 7
350 DATA 142, 1, 218, 159, 122, 126,
 164, 166, 126, 140
360 DATA 241, 83, 69, 65, 82, 67, 72
, 73, 78, 71
```

6809E

THE MAGIC NUMBER

by Michael Monck

Hello once again to all the coco users out there. I have changed my mind about doing this article about the different numbering system, this months article is going to be on addressing in memory. So here it is!

People have often asked me why the computer stores everything as numbers. This is because that is all it can do!.

This is usually what I tell them.

If I wanted to write a letter to you, where would I send it?
In other words, what address would I send it to.

For example:- I wish to send a letter to my friend Yanni Dellaportas who lives at the address 59 McCloud road in Mount Martha Victoria. The letter head would read:-

YANNI DELLAPORTAS
59 McCLOUD ROAD
Mt MARTHA

The computer, like my friend, has many addresses with certain things that "live there"

For example:-Immagine a street that is 65525 houses long, and that every house has someone living in it. The name of the street is "Memory"

All the house numbers are on the letterboxes and are very easy to see from the road. The only way to find out who lives in the house is to take a peek in the front door. The basic statement 'PEEK' does exactly that. It looks inside the memory address and tells you what, or who lives there. In other words it tells you what resides in that paticular memory address.

There are three main addresses. These are as follows
(1) Permanant addresses (such as ROM addresses)
(2) Static addresses (such as RAM addresses)
(3) Control addresses (such as soft switches)

All these addresses can be peeked at from basic or machine code. Here is how to do it.

BASIC-
Example:-We wish to see who resides at memory location 1024

```
10 PRINT PEEK(1024)
```

MACHINE CODE-

```
0100 START    LDA    1024T    ;LOAD A 1024D
0200 LOOP     JMP    LOOP     ;LOOP HERE
0300          END
```

(To examine the contents, set BRKPT to LOOP and, after executing the M/L program, examine the registers)

As you can see, basic is the much simpler way to examine memory, so all other examples will be given in basic.

In running either of the programs, you have peeked into the door of the house that is at 1024 MEMORY street. Make a note of who lived there.

Now clear the screen and type a line of 'A'. Then press enter until the line of 'A' is at the top of the screen. Then type 'RUN'

Has the number changed from the number that used to live in that address? if so, why? (In fact the number should now be the number 65)

If you had typed 'B' instead of 'A' then the numbers would have been 66

We can now assume the location 1024 is STATIC because it is possable to change. Type this next basic program in and run it.

```
10 PRINT PEEK(&HC000)
```

Now do as we did before with the 'A' and run the program again. Even go as far as switching the computer off and trying again. The result should be constant.

We can now assume the location &HC000 is a PERMANANT memory address.

This only leaves us with the CONTROL address.

The most common of these control addresses is the toggle soft switch. All we have to do to operate this switch is to peek at that address. That's all there is to a control address. Some controll addresses, however, require a number to be put in them. (POKEd into them) Lets take the speed up poke for example.

```
10 POKE 65495,0
```

This is an address to toggle the clock speed switch to the ON position. To turn this soft switch off again we must put a zero in another control address, thus-

```
10 POKE 65494,0
```

Will cause the computer to resume normal clock speed.

Well that is all I have for you this month. I hope I have cleared some things up for some people.

FROM:
MICHAEL MONCK
C-o BAYNE & TREMBATH
3 BONEO ROAD
ROSEBUD VIC 3940
TEL 059-86 8288

P.S. I can type better now that my hand is healed!

P.P.S. Here is a little computer joke-
1100100101011101010101010101011101010111011010101010101010101
0101011110101010010010101010101010100101001010101010101
0101010101010010010101010101010 ha ha ha

ASSEMBLY FILE

BY KEVIN

Let us take you on a voyage of discovery to explore the hidden world inside the CoCo's memory microchips.

The term MEMORY MAP is often referred to but seldom explained, so perhaps its time we made an attemt to explore our own CoCo's memory map.

Michael Monck has broadly described in his article the three types of memory address. Firstly there is the ordinary old RAM we have all come to know and desire and of course there is the ROM which makes CoCo do all those magic things. But seldomn explained are the control addresses suituated right at the top of the memory map.

Briefly the memory map is really only a table, mapping the allocation of individual memory addresses through the entire range of the computers memory. So lets look first at an overview memory map of a CoCo with the EDTASM+ cartridge plugged into the expansion port. Please note that here, the trouble you took to develop a little understanding of the hexadecimal numbering system pays dividends as most memory boundries fit neatly into the system.

ADDRESS	CONTENTS
$0000	Bottom of memory and start of RAM
$03FF	Top of RAM allocated to system use
$0400	Start of video text screen memory
$05FF	End of video text screen memory
$0600	Beginning of user (thats you) RAM
$3FFF	Top of RAM -16K systems
$7FFF	Top of RAM -32K systems
$8000	Start of EXTENDED BASIC ROM
$9FFF	End of EXTENDED BASIC ROM
$A000	Start of COLOR BASIC ROM
$BFF2	Duplicate immage of Interupt Vectors
$BFFF	End of COLOR BASIC ROM
$C000	Start of EDTASM+ ROM
$DFFF	End of EDTASM+ ROM
$FF00	Start of Input Output control addresses
$FFBF	End of Input Output control addresses
$FFC0	Sart of SAM control addresses
$FFDF	End of SAM control addresses
$FFF2	Start of Interupt Vectors
$FFFF	Top of memory

If you examine this list you will find two big gaps. The first of these lies from $4000 to $7FFF in a 16K system and is simply explained by the lack of the physical hardware (64K RAM chips) recuired to make use of this address range. The second gap lies from $E000 to $FEFF. Again we face a lack of physical hardware but making use of this gap is a little more obscure than simply plugging in RAM. Note too that if the EDTASM+ cartridge is removed then our gap starts at $C000 instead of $E000.

The hardware configuration of the CoCo includes a bank of 8 RAM chips to give you either 16K, 32K or 64K of hardware RAM; two 8K ROMs, one refered to as EXTENDED BASIC and the other COLOR BASIC; the CARTRIDGE PORT; the Peripheral Interface Adapters (PIA's) and finally the SYNCRONOUS ADDRESS MULTIPLEXER, otherwise known as the SAM chip. All these components are spoken to via there pre-allocated adresses as listed above.

When CoCo is cold started the Program Counter register in the CPU is loaded with the contents of bytes $BFFE and $BFFF. These. bytes contain the value $A027 which just happens to be the address where the machine language program we call COLOR BASIC begins EXECution.

CoCo then enters its RESET routine and if upon looking at address 113 (yes I know I'm using a decimal address - just happens to be the number I remembered) and finding zero (or at least no $55) will enter the cold start reset routine. Perhaps you now understand why the command line:

```
POKE113,0:EXEC&HA027
```

performs a cold start.

The SAM chip acts somthing like a set of traffic lights ensuring the software is always talking to the right hardware components. That is an oversimplification but will do for now. We or COLOR BASIC can manipulate the SAM chip via addresses $FFC0 through to $FFDF. POKEing any value to an odd numbered SAM address will SET the SAM register tied to that address. POKEing any value to the EVEN numbered address immediately below will CLEAR that register. In this manner we can choose the way in which we will tie our systems hardware (more on this later) to the 64K range of directly addressable memory. Perhaps the best example of the way in which we manipulate the SAM chip is the good old speed-up POKE.

```
POKE &HFFD7,0   SETS CPU rate control register (fast)
POKE &HFFD6,0   CLEARS CPU rate control register (normal)
```

The PIA's control communication between the computers heart and all of its attachments. Attachments includes the Keyboard, TV display, Cassette Player, Cartridge Port, Printer and whatever else the rest of the world can dream up. Each PIA uses 4 addresses to control the manner in which it carries out its INPUT/OUTPUT functions. Unlike the SAM chip the PIA is set up by writing specific values to its odd numbered addresses to determine how the data transmitted via the even numbered addresses is to be manipulated.

Naturally enough CoCo is most happy when all of these functions of the SAM and PIA's work in harmony with what it thinks it is doing. Obviously then, one of the first things the RESET routine does is set up. the PIA's and SAM chips.

BUT we have a long way to go yet before CoCo can hand control back over to you.

Further functions of the RESET routine are determining the amount of RAM installed in your computer, what level of ROM- ECB, DISK or ROMPACK CARTRIDGE and finally reserve itself a block of RAM for its own use.

CoCo when operating under BASIC needs a place to store its own variables such as the printer baud rate, location of the start of the users BASIC program, EXEC address of a machine language program. The data stored within this area of system RAM from $0000 to $3FF is considerable and when we know how provides us with an area where we can manipulate the BASIC ROM's. Perhaps a good example is the illegal PCLEAR0 command. When we POKE25,6:NEW we are manipulating our knowledge of the way BASIC works to give us more User RAM.

But we must yet consume even more of our precious RAM, this time to provide somewhere to store the ASCII numbers which can be read by the Video Display Generator and used to display our screen. The block of memory normally reserved for this purpose is that lying between $0400 and $05FF. This can be locted anywhere in memory however by simply reseting some of the registers in the SAM chip. Tandy chose this position for reasons of their own, I immagine the most likely reason being to avoid fragmentatipon of user RAM in a way compatible with all versions of CoCo.

Now for the most confusing feature of our memory map. All that I have explained so far relates to memory configured to MAP TYPE 0. Remember those cursed 64K RAM chips that only gave you an extra 16K of RAM. Do not despair! When operating with machine language we can make that unused 32K of RAM work for us. Controlling the SAM registers addressed at $FFDE and $FFDF causes CoCo to either look and the 64K range of addressable memory in the manner you are used to or alternatively all of the hardware addressed from $8000 to $FEFF (ie. all of the ROM's) may be switched off and any addressing within this range is directed by the SAM chip to the unused portion of those 64K RAM chips. But thats not all the SAM chip can do with those 64K RAM's. We can also page select our lower bank of RAM, addresses $0000 through to $7FFF, by directing any addressing in this range to either the top 32K or bottom 32K bank of our 64K RAM's depending on the way we set up the SAM control register addressed at $FFD4 and $FFD5. I guess I shouldn't need to tell you that when entering either MAP TYPE 1 or RAM PAGE 1 you become responsible for ensuring the Program Counter register continues to point to a sensible machine routine within the new blocks of RAM when reading those addresses.

And finally, let me make you aware of a trap many novice ML. programmers fall into once they begin to find their way around the ROM's. We have shown you in previous articles how you can EXEC certain routines within ROM as if they were separate ML. programs. For example if I type in EXEC &HA910 on my CoCo the screen will clear. But this will only work on those CoCo's using the same revision of the COLOR BASIC ROM in my machine. Anyone with one of the later CoCo's will simply crash the computer and need to RESET. There are certain addresses that TANDY guarantees will be common to all versions of ROM so all is not lost. At this level we can achieve some measure of compatibility between one anothers CoCo. Incidentally Software Spectrum sell dissasemblies of each of the ROM's. They are titled COLOR BASIC UNRAVELLED, EXTENDED COLOR BASIC UNRAVELLED and DISK BASIC UNRAVELLED. I think different revisions are published for each ROM revision.

FORTH

FORTH FORUM

SIMULATION

by John Poxon

Last month I introduced the equation:

```
Q=(11*PI/3)+6*(d*SQRT(1-d*d)+ARC
SIN(d))-PI*((d*d*d/3)-d)
```

Where Q=volume of water in a boiler (presumed, for the time being to be cold), PI=a constant and d=deviation from midline (centre-line) of liquid level; i.e. a deviation of 0 means that the tank is half full, a deviation of 1.0 means that it is full, and a deviation of -1.0 means that it is empty. (units are metres).

I remarked that the Newton-Rapson method, an iteritive technique, can be used to find the deviation given a known volume of liquid in the boiler, if the equation is set equal to zero. i.e., if

```
-Q+(11*PI/3)+6*(d*SQRT(1-d*d)+AR
CSIN(d))-PI*((d*d*d/3)-d)=0      (EQU 1)
```

The Newton-Rapson method is actualized in the equation:-

```
d2=d1-(f(d1))/f'(d1))     (EQU 2)
```

Where d2=new value of guess and d1=old value of guess (of deviation).

f=EQU 1 (above) and f'=first derivative of f.

Note that EQU 2 must function inside a loop, wherein d2 is continually recalculated in terms of d1, and d1 successively reset to the value of d2. If this is not clear, look at the programs in last months Rainbow, where EQU 2 was set out in the Pascal lines:-

```
14 V1:=11.519-Q+6*(d1*SQRT(1-d
   1*d1)+ARCSIND1)-3.142*(((d1
   *d1*d1)/3)-d1);
15 V2:=3.142*(1-d1*d1)+((-12*d
   1*d1*d1+12)/(SQRT(1-d1*d1)));
16 d2:=d1-(V1/V2);
```

The intention of this article is to represent the boiler

model by a FORTH program. Though the FORTH program appears much different to the Pascal and BASIC programs, it achieves approximately the same effect.

Before we get on with finding the deviation, I attract your attention to the constant PI, the variables D1, D2, N and Q, and the need to evaluate square roots and arcsin. Pause a moment to reflect on how you would calculate the latter functions if they were not resident in your computer ROM or RAM, and on how nice it would be to have sections of your program read as an english description of the behaviour of that section of the program.

For the moment (I know you're aching to do it anyway), type in the program listing exactly as presented below. If you're using John Redmonds A*FORTH, and you're uncertain how to "get it in", refer to the instructions on using the A*FORTH editor presented in last months article.

```
 0 VARIABLE D1
 0 VARIABLE D2
 0 VARIABLE N
 0 VARIABLE SUMTERMS
 0 VARIABLE Q
 : *PI 355 113 */ ;
 : 1-D1*D1 10000 D1 @ 10 / DUP * - 10 / ;
 : SQRT DUP 0 > NOT IF DROP 0 LEAVE THEN DUP 2/ 2/  2/  2/
1+ BEGIN 2DUP /
   2DUP - ABS 1 > WHILE + 2/ REPEAT ROT 2DROP ;
 : SQRT(1-D1*D1) 1-D1*D1 10 * SQRT 10 * ;
 : D1*ROOT D1 @ SQRT(1-D1*D1) 1000 */ ;
 : D1EN/N D1 @ N @ 1- 0 DO D1 @ 1000 */ LOOP N @ / ;
 : FACTOR 1000 N @ 1- 1 DO I * I 1+ / 2 +LOOP ;
 : TERM FACTOR D1EN/N 1000 */ ;
 :  ARCSIND1  D1 @ SUMTERMS ! 80 3 DO I N ! TERM DUP 0= IF
LEAVE THEN
   SUMTERMS +! 2 +LOOP ;
 : D1E3 D1 @ DUP DUP 1000 */ 1000 */ ;
 : (D1E3/3)-D1 D1E3 3 / D1 @ - ;
 : LASTOFV1 (D1E3/3)-D1 *PI ;
 : V1 Q @ NEGATE 11519 + D1*ROOT ARCSIND1 SUMTERMS @ + 6 *
+ LASTOFV1 - ;
 : 2NDTM 12000 NEGATE D1E3 M* 12000 M+ SQRT(1-D1*D1) M/ ;
 : V2 1-D1*D1 *PI 2NDTM + ;
 : D22 D1 @ V1 1000 V2 */ - DUP D2 ! ;
 : BOILER Q ! 40 0 DO D22 D1 @ - ABS 15 < IF ." DONE! " CR
CR  ."
   LEVEL DEVIATION = " D2 @ . ." M.METRE." LEAVE THEN D2 @
D1 ! LOOP ;
```

"RUN" it by leaving the editor (after compiling, of course), and typing in the following, for example:

11519 BOILER (RETURN) (i.e. multiply the volume by 1000 before entering)

After a very short period of time the program will return with a level deviation from midline (in this case 0, since 11.519 cubic metres is half the volume of the boiler).

Try other values of volume. Note that the deviation returned is in mm. I suggest that you restrict your early experiments to values of volume between about a quarter to three quarters of 23 cubic metres, since:

a) large deviations of level take longer to calculate, and

b) an instability exists at low volume levels which causes cycling of the D1, D2 calulation within BOILER.

I am inclined to believe that the instability is caused by my use of single length calculations and subsequent rounding errors (rather than a mathematical error). I will investigate the instability later when I have more time, but let me hasten to say that such an instability is not a problem, since a real boiler would be controlled to operate very close to midline. In the event that the volume of water contained in a real boiler were to run as low as the instability level in this model I would recommend seeking a shrapnel shelter! On the other hand if a boiler ran excessively full, an imminent danger of water entering the boiler superheater tubes would demand emergency corrective action. The real boiler which served as a basis for this model is, in fact, controlled to within plus or minus 25 mm.

Now that you have had a play with it, we can discuss it a little.

The variables have the following functions: D1=initial guess, D2=newly calculated guess, N=the value of the index in the calculation of successive terms of ARCCSIND1, SUMTERMS contains the sum of the values thus calculated, and Q=the operator inputted value of tank contents. When the model is developed further, the value of Q value will be generated by other parts of the program.

I must acknowledge the use of John Redmonds previously published SQRT routine. There seemed little point in re-inventing the wheel in this case, especially since illness and life in general have severely cut short the time available for preparation of this article.

I suggest that you carefully compare the boiler model equation with the words in BOILER, and work backwards through defining words to establish the actual functions. Examination of most words will reveal that previously defined words are employed, thereby creating a structured format.

There are various ways in which trigonometrical or other functions can be evaluated, including:

a) using a software routine.

b) using a "look-up table" (which John intends to deal with in a future article).

c) having a (mathematics) dedicated microprocessor running in parallel to the one which is running the program.

d) using an analogue computer.

When selecting a method to find the value of arcsin (d1) I chose the first option. Why? Largely because I could'nt bear the thought of typing in the 1000 elements of the look-up table I would need, so I decided to do it via a recursive routine ARCSIND1.

continued on P 41

FRICKER'S FOLLIES

by Jack Fricker

In this series I am going to try to explain how to use BASIC09, try to help you convert your Colour BASIC programs over to BASIC09 and hopefully to write new BASIC09 programs.

To start with we will look at some very simple programs like those that you started with when you first got your CoCo. Things like a for next/loop and printing your name on the screen like the simple ones listed here.

```
 10 FOR A=1 TO 10
 20 PRINT "HELLO FRED"
 30 NEXT A
```

Or the BASIC09 version.

```
 PROCEDURE Hello
 for a=1 to 10
 print "hello fred"
 next a
```

Now there are a couple of things that may be new to you.

The first one is the word PROCEDURE. OS9 requires that every procedure has a name. The name of this little offering is "hello".
This name is also used when the program is "packed" or when it is "run".

Once Packed, programs can be run from OS9 command mode by just typing the name of the procedure, just like any other OS9 program. In this case by typing hello. To be able to do this RUNB must be your current commands directory.

The next thing you will notice is that I left the line numbers out. I did'nt really leave them out because BASIC09 doesn't require them to work.
I could have typed them and the program would have worked just the same. There are some circumstances that require line numbers to be used like GOTO or GOSUB.

You can of course use line numbers if they make you more comfortable. Remember that you DON'T have to number EVERY line. The way I do it I just number ones that are referenced by GOTO's or GOSUB's or a remark to mark the start of a routine.

The next thing you may notice is that the lines are typed in lower case. BASIC09 will automatically change lower case words to upper where the word is a BASIC09 reserved word.

BASIC09 will also pretty print a]listing to show you the control structure as in the next example where the lower case has been changed by BASIC09 but the address bytes that BASIC09 inserts have been left out.

I will not be touching on the addresses this month, but I will cover them another time.

```
 PROCEDURE hello
 FOR a=1 TO 10
   PRINT "hello"
 NEXT a
```

Pretty printing is the indentation of the code within the control loops.

More than one command can be put on one line just the same as in Colour BASIC. The syntax for this varies from the Colour BASIC seperator. The Colour BASIC one is the colon : where the back slash \ is required under BASIC09.

```
 Colour BASIC
 FOR A=1 TO:PRINT"HELLO":NEXT

 BASIC09:
 PROCEDURE hello
 for a=1 to 10 print"hello" next a
```

I might mention that this form is frowned upon in BASIC09 and other structured languages because it makes programs harder to read and unlike Colour BASIC it does NOT make the program run any faster or save any memory.

You may also notice that in the last listing for Colour BASIC I did not need to add the A after next, but I did under BASIC09. BASIC09 is a much more strict form that Colour BASIC but it does have the advantage of running very much faster.

NOW! is the time to subscribe to Australian RAINBOW

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

RATES

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

BACK ISSUES

MiCo —
First Issue Oct '83
□
$3/copy

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

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

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

DISKS & TAPES

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

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

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

BOOKS

Byte
SORRY
OUT OF STOCK

Help
Medium $9.95 □

Facts
Advanced $11.95 □

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

PLUS

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

BULLETIN BOARD

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

□□□□□□□□□□□□□□

□ VISA BANKCARD □
MASTERCARD □

______________ Signature

□ CASH
□ CHEQUE
□ MONEY ORDER

VISA

Authorised
Amount $ ______________

BLOCK CAPITALS
PLEASE

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

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

FIRST NAME SECOND NAME

Address

PC

Telephone Number STD Code Local Number

New Subscription □ Renewal □

ADVENTURE TAPE #1

THE TOMB OF TUTANKHAMEN
UNDERWATER ADVENTURE
THE LOST CITY OF JENDUSA
SPACESTATION SPRATOS
RESCUE THE PRINCESS SHARINE
DUNGEONS OF SKELOS
HAUNTED MANSION
THE LOST TREASURE OF INCA

Requires 32K, Extended
Color Basic.
Only $14.95 post paid.
For disk version,
add extra $3.00

Nick Cooper
80 Swaine Ave.,
Toorak Gardens,
S.A. 5065.
Phone (08) 3325982

DLT002 CoCo — INTERFACE

•Connect your CoCo to the outside world •Controlling robot arms, turtle robots, electrical equipment, home alarms...up to 32 devices!!
•Can be programmed to sense external devices such as temperature, humidity (with appropriate electronic circuit)
•Program manual included and even a circuit for you to control relay.

*** ONLY $95.00 ***

available from: DAVID LY,
8 AWARBA AVE.,
WAGGA NSW 2650

(Please allow 6 weeks for delivery)

BUDGET SOFTWARE

FREEPOST NO 2. 5 BANKSIA RD.
KELMSCOTT. PERTH.WA.6111.

><><><><><><><><><><><><><><><><><

NEW NEW NEW NEW NEW NEW NEW.NEW.
BIZZ WIZZ
written for the small businessman eg:- electrician,plumber,Shops, carpenter,window cleaner,etc. The program is DISK BASED.and very well documented.. BIZZ WIZZ will handle all of your invoicing, accounting, pricing, includes checkbook feature,typewriter function etc...TOTALLY USER FRIENDLY. price is a low $59.95 inc post. please stipulate printer name.

================================

NEW NEW NEW NEW NEW NEW NEW NEW
HINTS AND TIPS FOR THE
COCO ADVENTURER
Includes hints,tips & solutions to over 25 adventures inc:- Sanctum.Seaquest.Sands of Egypt. Dallas.Shennan.H/House etc etc. price $10.00 inc postage......
BANKCARD & MASTERCARD WELCOME.
BUDGET SOFTWARE (09) 390 5277.

================================

SAVE $ $ $ $

- CPA-80(s) 25cm, 100cps, Tractor friction feed, Front Dip Sw for options, Sockets for 4K buffer (6116), Superior Square Pin Head Technology. RS 232 I/F. Ideal for CoCo Graphics and more. $321.17
- CPB-80(s) 25cm, 130cps, 2K Buffer fitted. Graphics enhanced CPA-80. RS232 I/F. $336.24
- CPB-136(s) 38cm, 130cps, 2K Buffer fitted (exp to 4K). All CPA-80 features and more. Linear Tractor and friction paper feeder. $542.97

Centronics I/F versions also available at lower cost.

- Disk Drive System for CoCo, includes: Professional moulded case for interface card. All drive cables. Precision NC case with internal PS. single 40TK DS DD, drive expandable to 2 drives. $399 inc.

Special dealer and club prices available. Call for details. All prices tax and delivery to major centres. We import directly from the Japanese maker and stock service manuals and spare parts for all products we offer

energy CONTROL

ENERGY CONTROL INTERNATIONAL PTY. LTD.
P.O. Box 6502, Goodna Qld 4300
Brisbane AUSTRALIA
Phone (07) 288 2455
Telex AA43778 ENECON
P.O. Box 12153, Wellington North,
NEW ZEALAND
Phone (4) 72 6467 TLX NZ 30135

Prices subject to change without notice.

MK1 SERIAL/PARALLEL PRINTER INTERFACE

CONNECT CO-CO I or II to a PARALLEL PRINTER
Revised MK1 PRINTER INTERFACE
Features:
* EXTRA SERIAL PORT for MODEM,no more plugging /unplugging cables.
* Compatible with Standard Centronics Parallel Printers eg:EPSON,GEMINI,BMC,CP80,TANDY ETC.
* Plugs into CO-CO or CO-CO II Serial Port and includes all cables and connectors.
* SIX Switch selectable Baud rates 300,600,1200,2400,4800,9600
* Power Pack is required for Printers not supplying power at pin 18 on the Parallel Connector eg:EPSON,BMC,CP80.
* Increases Printing Speed by up to 30% on TANDY DMP100/200 Printers

ONLY : $94.95 (including postage)
Add $9 for Power Pack if required

AVAILABLE FROM: G.& G. FIALA
P.O. BOX 46
THORNLEIGH. NSW.2120
phone:(02)-84-3172

user group CONTACTS

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

ADELAIDE	JOHN HAINES	08 278 3560
ADELAIDE NTH	STVN EISENBERG	08 250 6214
ALBURY	RON DUNCAN	060 43 1031
ARMIDALE	TOM STUART	067 72 8162
BAIRNSDALE	COLIN LEHMANN	051 57 1545
BALLARAT	MARK BEVELANDER	053 32 6733
BANKSTOWN	CARL STERN	02 646 3619
BLACKTOWN	KEITH GALLAGHER	02-627-4627
BLACKWATER	ANNIE MEIJER	079.82.6931
BLAXLAND	BRUCE SULLIVAN	047 39 3903
BOWEN	TONY EVANS	077 86 2220
BRASSALL	BOB UNSWORTH	07 201 8659
BRIGHTON	GLENN DAVIES	08 296 7477
BRISBANE EAST	ROB THOMPSON	07 848 5512
BRISBANE SH	PATRICK SIMONIS	07 209 3177
BRISBANE SW	GRAHAM BUTCHER	07 376 3400
BRISBANE WEST	BRIAN DOUGAN	07 30 2072
BROKEN HILL	DEAN PARADICE	080 6701
BUNBURY Call Tandy for local details.		
BUNDABERG	JIM McPHERSON	071 72 8329
CAMBERWELL	TONY BALDWIN	03 728 3676
CAMDEN	KEVIN WINTERS	046.66.8068
CAMPBELLTOWN	LEO GINLEY	02 605 4572
CANBERRA NTH	JOHN BURGER	062 58 3924
CANBERRA STH	LES THURBON	062 88 9226
CAULFIELD	JEFF SHEEN	03 528 3724
CHATSWOOD	BILL O'DONNELL	02 411 3336
' or	MARK ROTHWELL	02 817 4627
CHURCHILL	GEOFF SPOWART	051 22 1389
COLYTON TEENS	DWAYNE MANSON	02 623 5805
COOMA	ROSS PRATT	0648 23 065
COORANBONG	GEORGE SAVAGE	049 77 1054
DANDENONG	DAVID HORROCKS	03 793 5157
DARWIN	BRENTON PRIOR	089.81.7766
DENILIQUIN	WAYNE PATTERSON	058 81 3014
DONCASTER	JUSTIN LIPTON	03 857 5149
DUBBO	GRAEME CLARKE	068 89 2095
FORBES	JOHANNA VAGG	068 52 2943
FORSTER	GARY BAILEY	065 54 5029
FRANKSTON	BOB HAYTER	03.783.9748
GIPPSLAND STH	PAT KERMODE	056 74 4583
GLADSTONE	ALBERT VAN GORKUM	079 72 2353
GOLD COAST	SHERYL BENTICK	075-39-2003
GOSFORD	PETER SEIFERT	043 32 7874
GOULBURN VALLEY	TONY HILLIS	058 59 2251
GRAFTON	DAVID HULME	066.42.0627
GREENACRES	BETTY LITTLE	08 261 4083
HASTINGS	MICHEAL MONCK	059.86.8288
HERVEY BAY	LESLEY HORWOOD	071 22 4989
HILLS DIST	DENNIS CONROY	02 671 4065
HOBART	BOB DELBOURGO	002 25 3896
HORNSBY	ATHALIE SMART	02 848 8830
IPSWICH	MILTON ROWE	07 281 4059
JUNEE	PAUL MALONEY	069 24 1860
KALGOORLIE	TERRY BURNETT	090.21.5212
KENMORE	GRAHAM BUTCHER	07 376 3400
LEETON	CHRIS NAGLE	069 53 2969
LITHGOW	DAVID BERGER	063 52 2282
LIVERPOOL	LEONIE DUGGAN	02-607-3791
MACKAY	LEN MALONEY	079511333x782
MACLEOD	ROBIN ZIUKELIS	03 450211x465
MacQUARIEFIELDS	KIETH ROACH	02 618 2858
MAFFRA	MAX HUCKERBY	051 45 4315
MAITLAND	LYN DAWSON	049 49 8144
MARYBOROUGH	NORM WINN	071 21 6638
MELBOURNE	JEFF SHEEN	03 528 3724
MELTON	MARIO GERADA	03 743 1323
MILDURA	SCOTT HOWISON	050 23 6016
MORPHETTVALE	KEN RICHARDS	08 384 4503
MOREE	ALF BATE	067 52 2465
MORWELL	GEORGE FRANCIS	051 34 5175
MT ISA	PAUL BOUCKLEY-SIMONS	077 43 6280
MUDGEE	BRIAN STONE	063-72-1958
MURGON	PETER ANGEL	071 68 1628
NAMBUCCA HDS	WENDY PETERSON	065 68 6723
NEWCASTLE	LYN DAWSON	049 49 8144
NOWRA	ROY LOPEZ	044 48 7031
ORANGE	STEVE LOVETT	063.62.4025
' or	JIM JAMES	063 62 8625
PARKES	DAVID SMALL	068 62 2682
PENRITH	ALEX SCHOFIELD	047 31 5303
PERTH	IAN MACLEOD	09 448 2136
PORT LINCOLN	JOHN BOARDMAN	086 82 2385
PORT MacQUARIE	RON LALOR	065 83 8223
PORT NOARLUNGA	ROB DALZELL	08 386 1647
PORT PIRIE	KEVIN GOWAN	086 32 1368
RINGWOOD	ANDREW RAWLINGS	03 700 4498
ROCKHAMPTON	KEIRAN SIMPSON	079 28 6162
ROSEVILLE,	KEN UZZELL	02 467 1619
SALE	BRYAN McHUGH	051 44 4792
SANDGATE	MARK MIGHELL	07 269 5090
SCARBOROUGH	PETER MAY	07 203 6723
SEACOMBE HTS	GLENN DAVIS	08 296 7477
SHEPPARTON	ROSS FARRAR	058 25 1007
SMYTHESDALE	TONY PATTERSON	053 42 8815
SPRINGWOOD	DAVID SEAMONS	047 51 2107
STURT	MARY DAVIS	08 296 7477
SUNBURY	JACK SMIT	03.744.1355
SUTHERLAND	IAN ANNABEL	02 528 3391
SWAN HILL	BARRIE GERRAND	050.32.2838
SYDNEY TEENS	ROD HOSKINSON	02 48 5948
SYDNEY EAST	JACKY COCKINOS	02 344 9111
TAMWORTH	ROBERT WEBB	067 65 7256
TAHMOOR	GARY SYLVESTER	046 81 9318
TONGALLA	TONY HILLIS	058 59 2251
TOOWOOMBA	GRAHAM BURGESS	076 30 4254
TOWNSVILLE	JOHN O'CALLAGHAN	077 73 2064
TRARALGON	MORRIS GRADY	051 66 1331
UPPER HUNTER	TERRY GRAVOLIN	065 45 1698
WAGGA WAGGA	CES JENKINSON	069 25 2263
WHYALLA NORRIE	CHRIS HUNTER	086 45 3395
WONTHAGGI	PAT KERMODE	056 74 4583
WYNYARD	ANDREW WYLLIE	002 35 1839
YARRAWONGA	KEN SPONG	057 44 1488
SPECIAL INTEREST GROUPS		
' BUSINESS		
BRIZBIZ	BRIAN BERE-STREETER	07 349 4696
TOP - TELEWRITER, DYNACALC PROCOLOR		
BRISBANE	GEOFF TOLPUTT	07 44 6084
' OS9 GROUPS		
BANKSTOWN	CARL STERN	02 646 3619
BRISBANE	JACK FRICKER	07 262 8869
KALGOORLIE	TERRY BURNETT	090.21.5212
MONARO	FRED BISSELING	0648 23263
PENRITH	BOB THOMSON	047 30 2468
SYDNEY EAST	JACKY COCKINOS	02.344.9111
SYDNEY NTH	MARK ROTHWELL	02 817 4627
BLAXLAND 128K	BOB THOMSON	047 30 2468
' MiCo GROUPS		
CARLISLE	STUART HALL	08 361 1922
LITHGOW	DAVID BERGER	063 52 2282
PORT LINCOLN	JOHN BOARDMAN	086 82 2385
ROCKHAMPTON	TIM SHANK	079 28 1846
SYDNEY	RAJA VIJAY	02 519 4106
' TANDY 1000 / MS DOS		
BRISBANE	BRIAN DOUGAN	07 30 2072
SYDNEY	ROGER RUTHEN	047.39.3903
' FORTH		
PORT LINCOLN	JOHN BOARDMAN	086 82 2385

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

AUSTRAILIAN COCO

THIS MONTH

- * Letters
- * Hangman
- * Quick on the Draw
- * Narzod
- * Frograce
- * Impossible
- * And lots more for you to enjoy